[MS-PUBWS]:

Publishing Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial Availability
6/27/2008	1.0	Major	Revised and edited the technical content
10/6/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Minor	Updated the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Minor	Updated the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	No Change	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	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	2.04	No Change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	No Change	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.1	Minor	Clarified the meaning of the technical content.
2/11/2013	3.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.2	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
11/18/2013	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.3	Minor	Clarified the meaning of the technical content.
7/31/2014	3.4	Minor	Clarified the meaning of the technical content.
8/24/2015	4.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Publishing Web Service Protocol enables a protocol client to perform tasks related to template-based published content on a protocol server. The protocol client receives information about one or more objects from the protocol server, and the protocol client submits to the protocol server changes to the states of one or more objects.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

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

content placeholder: A region within a page layout that is populated dynamically with the value of the publishing page field to which it is bound.

content slice: A container of sites that have friendly URLs.

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

content type identifier: A unique identifier that is assigned to a **content type**.

deployment package: A collection of files that represent a serialized snapshot of data. A deployment package is stored as **XML** files that describe the deployment objects and their relationships, and a binary file for each object. Optionally, the resulting set of files can be compressed into one or more files in the compressed PRIME data format (CMP).

descendant content type: Any **content type** that inherits settings from another content type.

displayed version: Document version information that is formatted for display in the user interface. The displayed version uses the format MajorVersion.MinorVersion, where MajorVersion is the published version number and MinorVersion is the draft version number, separated by a decimal point. See also **major version** and **minor version**.

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

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

document stream: A byte stream that is associated with a document, such as the content of a file. Some documents do not have document streams.

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

file: A single, discrete unit of content.

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

friendly URL: A web address that is easy for users to read and for search engines to crawl.

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

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

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

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

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

major version: An iteration of a software component, document, or list item that is ready for a larger group to see, or has changed significantly from the previous major version. For an item on a SharePoint site, the **minor version** is always "0" (zero) for a major version.

minor version: An iteration of a software component, document, or list item that is in progress or has changed only slightly from the previous version. For an item on a SharePoint site, the minor version number is never "0" (zero) and is incremented for each new version of an item, unless a **major version** is explicitly published. When minor versioning is disabled on a SharePoint site, only major version numbers are incremented, and the minor version is always "0" (zero).

moderated object: An object for which a moderator reviews and either approves or rejects additions or changes to that object or to objects that are contained by that object. New objects and changes to existing objects can be seen by other users only after they have been approved by the moderator.

moderation status: A content approval status that indicates whether a list item was approved by a moderator.

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

page layout: A dynamic web template that is stored as a document. It contains content placeholders that bind to **fields** of a publishing page. A page layout has an associated content type that determines which publishing pages it can be bound to.

publishing object: Any file, document, or list item that is versioned or moderated, or has a publishing schedule.

publishing page: A document that binds to a page layout to generate an HTML page for display to a reader. Publishing pages have specific fields that contain the content that is displayed in an HTML page.

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

site collection: A set of websites (1) that are in the same content database, have the same owner, and share administration settings. A site collection can be identified by a **GUID** or the

URL of the top-level site for the site collection. Each site collection contains a top-level site, can contain one or more subsites, and can have a shared navigational structure.

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

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

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

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

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

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

source variation site: A website (2) that contains a collection of publishing pages to be copied to other sites, which are referred to as target variation sites. After the publishing pages are copied to a target variation site, they can be translated into another language. See also **target variation site**.

target variation site: A website (2) to which a collection of publishing pages were copied from another site, which is referred to as a source variation site. See also **source variation site**.

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

variations: An application that facilitates translation and related management processes for websites (2) and publishing pages. It can be used to copy content from one site, which is referred to as the source variation site, to one or more other sites, which are referred to as target variation sites. After the content is copied, it can be translated into different languages for those target variation sites.

web crawler: A search component that traverses websites, downloads content from those sites, and submits that content for indexing.

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

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

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

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

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

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

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

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

XMLHTTPRequest (XHR): A software component that is used by browser-based scripts to transfer data between a web browser and a web server.

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

1.2 References

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

1.2.1 Normative References

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

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

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

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

[RFC1766] Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, March 1995, http://www.ietf.org/rfc/rfc1766.txt

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

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

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

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

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

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

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

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

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

1.2.2 Informative References

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

1.3 Overview

This protocol provides publishing and content management-related functionality. Operations are exposed as **WSDL operations** that are initiated by the client. The protocol client sends a **SOAP message** to the protocol server to execute a specific operation using the supplied parameters. When the operation finishes execution, the protocol server sends a SOAP message that contains the optional output parameters to the protocol client. The Publishing Web Service protocol defines fifteen operations which are divided into five specific classes.

1.3.1 Page Layout Operations

This protocol exposes three operations associated with **page layout**. The protocol client can create page layouts, disconnect **publishing pages** from page layouts and reconnect publishing pages to page layouts.

1.3.2 Status Operations

This protocol exposes three operations which retrieve specific properties from a **publishing object** or set of publishing objects that are stored on the protocol server.

1.3.3 Translation Operations

This protocol exposes two operations that export and import multilingual content. These operations are associated with a process or processes that export, translate, and then import publishing pages associated with a **source variation site**. Content is exported and imported as a **deployment package**.

1.3.4 Wait Operation

This operation causes the protocol server to wait for a specific amount of time before sending an empty response to the protocol client.

1.3.5 Friendly URLs Operation

This protocol exposes six operations which obtain **sites** containing friendly URLs, friendly URLs or properties associated with a single friendly URL.

1.4 Relationship to Other Protocols

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

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

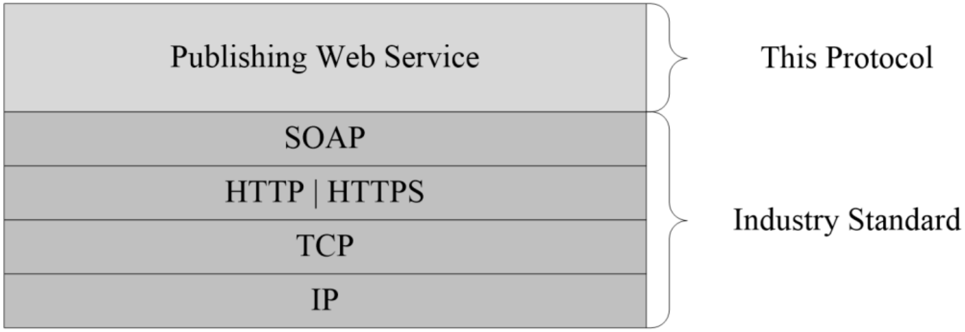

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

This protocol is used in several independent scenarios associated with the classes of operations described in section 1.3.

The Page Layout operations are used by a remote management suite to create new page layouts and to disconnect publishing pages from page layouts and reconnect publishing pages to page layouts.

The Status operations provide remote access to versioning, moderation, and scheduling information to drive external workflows and for views that summarize information about publishing objects.

The Translation operations are used with an automated translation workflow.

The **Wait** operation is used in browser scripts by the synchronous **XMLHTTPRequest (XHR)** object to block the current thread of execution for a specific amount of time. Any compliant protocol client that adheres to this specification can use the operation.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports:** This protocol uses multiple transports with SOAP as described in section 2.1.

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 in [SOAP1.1] section 4 SOAP Envelope, or in [SOAP1.2/1] section 5 SOAP Message Construct. Protocol server faults MUST be sent to the protocol client using either HTTP status codes as specified in [RFC2616] section 10, or using **SOAP faults** as specified in [SOAP1.1] section 4.4, or in [SOAP1.2/1] section 5.4.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and **WSDL**, as specified in [WSDL].

2.2.1 Namespaces

This section specifies **XML namespaces** using the mechanisms specified in [XMLNS]. Although each specific **XML namespace prefix** is uniquely associated with each XML namespace, the specific XML namespace prefix choice is implementation-specific and not significant for interoperability. The following table describes these namespaces.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/sharepoint/soap/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
(none)	http://schemas.microsoft.com/sharepoint/soap/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
ArrayOfFriendlyUrl	Collection of friendly URL objects in a site.
ArrayOfFriendlyUrlBasedWeb	A collection of FriendlyUrlBasedWeb objects.
ArrayOfPublishingObjectStatus	Encapsulates an array of **PublishingObjectStatus** elements.
ArrayOfString	Encapsulates an array of strings.
FriendlyUrl	Represents properties of a **FriendlyUrl** object.
FriendlyUrlBasedWeb	Represents properties of a site that contains friendly URLs.
PublishingObjectStatus	Encapsulates several properties related to the state and status of a publishing object.

2.2.4.1 PublishingObjectStatus

Represents properties of a publishing object.

```
<s:complexType name="PublishingObjectStatus">
  <s:sequence>
    <s:element name="ObjectType">
      <s:simpleType>
        <s:restriction base="s:string">
          <s:enumeration value="File"/>
          <s:enumeration value="AccessDenied"/>
          <s:enumeration value="FileNotFound"/>
          <s:enumeration value="UnrecoverableFailure"/>
          <s:enumeration value="Undefined"/>
        </s:restriction>
      </s:simpleType>
    </s:element>
    <s:element name="Url" type="s:string" minOccurs="0"/>
    <s:element name="PublishingUrl" type="s:string" minOccurs="0"/>
    <s:element name="Description" type="s:string" minOccurs="0"/>
    <s:element name="LastMajorVersion" type="s:decimal"/>
    <s:element name="LastMajorModifiedTime" type="s:dateTime"/>
    <s:element name="LastMinorVersion" type="s:decimal"/>
    <s:element name="LastMinorModifiedTime" type="s:dateTime"/>
    <s:element name="ScheduledStartTime" type="s:dateTime"/>
    <s:element name="ScheduledEndTime" type="s:dateTime"/>
    <s:element name="ModerationStatus">
      <s:simpleType>
        <s:restriction base="s:string">
          <s:enumeration value="Approved"/>
          <s:enumeration value="Denied"/>
          <s:enumeration value="Pending"/>
          <s:enumeration value="Draft"/>
          <s:enumeration value="Scheduled"/>
        </s:restriction>
      </s:simpleType>
    </s:element>
  </s:sequence>
</s:complexType>
```

ObjectType: An enumeration that identifies the success or failure of the operation. MUST be one of the following values:

Value	Meaning
File	The publishing object status properties were retrieved successfully, and the accompanying element values are valid.
FileNotFound	The specified publishing object does not exist or the current user is not authorized to see it.
AccessDenied	Access was denied while retrieving the status of the specified publishing object.
Undefined	The publishing object was specified by a **URL** with an unsupported scheme, the object specified is not a publishing object, or the publishing object was explicitly excluded from the result set. See section 3.1.4.8.2.1.
UnrecoverableFailure	Unrecoverable failure while attempting to fetch the status of the specified publishing object.

Url: The URL of the publishing object. If the **ObjectType** element is set to "File", this element MUST be present. If the **ObjectType** element is not set to File and this element is present, the protocol client MUST ignore it.

PublishingUrl: A URL used to retrieve a human-readable HTML representation of the publishing object. If the **ObjectType** element is set to File, this element MUST be present. If the publishing object is a publishing page, the value MUST be the URL of the page itself. Otherwise, the value MUST be any URL that can be used to obtain an HTML page that provides a human-readable view of the publishing object. If the **ObjectType** element is not set to File and this element is present, the protocol client MUST ignore it.

Description: A brief message explaining the failure in the case that the **ObjectType** element is set to UnrecoverableFailure. If the **ObjectType** element is not set to UnrecoverableFailure this element MUST not be present.

LastMajorVersion: The **displayed version** of the latest **major version** of the publishing object. The integral part of the decimal contains the major version number and the fractional part contains the minor version number. In the case of a major version, the major version number MUST be an integer between 1 and 8388608 and the minor version number MUST be 0. If the publishing object does not have a major version, this element MUST be set to 0. If the **ObjectType** element is not set to File, the protocol client MUST ignore this element.

LastMajorModifiedTime: The date and time at which the most recent major version of the publishing object was modified. If the publishing object has no major version, this element MUST be set to 0001-01-01T00:00:00. If the **ObjectType** element is not set to File, the protocol client MUST ignore this element.

LastMinorVersion: The displayed version of the latest **minor version** of the publishing object. The integral part of the decimal contains the major version number and the fractional part contains the minor version number. In the case of a minor version, the major version number MUST be an integer between 0 and 8388607 and the minor version number MUST be an integer between 1 and 511. If the publishing object has no minor version, this element MUST be set to 0. If the **ObjectType** element is not set to File, the protocol client MUST ignore this element.

LastMinorModifiedTime: The date and time at which the latest minor version of the publishing object was modified. If the publishing object has no minor version, this element MUST be set to 0001-

01-01T00:00:00. If the **ObjectType** element is not set to File, this element MUST be ignored by the protocol client.

ScheduledStartTime: The date and time at which the publishing object was or will be made available for viewing. If the container (the **list**, **document library**, or **folder**) of the publishing object does not support scheduling, this element MUST be set to 0001-01-01T00:00:00. If the container supports scheduling but the publishing object is set to be published immediately, the element MUST be set to 1900-01-01T00:00:00Z. If the **ObjectType** element is not set to File, this element MUST be ignored by the protocol client.

ScheduledEndTime: The date and time at which the publishing object was or will be made unavailable for viewing. If the container (the list, document library, or folder) of the publishing object does not support scheduling, this element MUST be set to 9999-12-31T23:59:59.9999999. If the container supports scheduling but the publishing object is set to never be un-published, this element MUST be set to 2050-01-01T00:00:00Z. If the **ObjectType** element is not set to File, this element MUST be ignored by the protocol client.

ModerationStatus: The current **moderation status** of the publishing object. MUST be one of the following values, each of which corresponds to one of the moderation status values specified in [MS-WSSFO3] section 2.2.1.2.13.

Value	Meaning
Approved	The publishing object is available for display in public views. Corresponds to a moderation status value of 0 as specified in [MS-WSSFO3] section 2.2.1.2.13.
Denied	The latest request for approval on the publishing object was denied. Corresponds to a moderation status value of 1 as specified in [MS-WSSFO3] section 2.2.1.2.13.
Pending	The latest request for approval on the publishing object is pending approval. Corresponds to a moderation status value of 2 as specified in [MS-WSSFO3] section 2.2.1.2.13.
Draft	The publishing object is currently being edited; no approval process is currently active on this item. Corresponds to a moderation status value of 3 as specified in [MS-WSSFO3] section 2.2.1.2.13
Scheduled	The publishing object has been approved for display and is scheduled to be automatically published at a specific time. Corresponds to a moderation status value of 4 as specified in [MS-WSSFO3] section 2.2.1.2.13

If the container (the list, document library, or folder) of the publishing object does not support **moderated objects**, this element MUST be set to Approved. If the **ObjectType** element is not set to File, this element MUST be ignored by the protocol client.

2.2.4.2 ArrayOfString

The **ArrayOfString** complex type represents a sequence of strings.

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

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

string: A single element in the string array.

2.2.4.3 ArrayOfPublishingObjectStatus

The **ArrayOfPublishingObjectStatus** complex type represents a sequence of **PublishingObjectStatus** elements.

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

PublishingObjectStatus: A single element in the **PublishingObjectStatus** array.

2.2.4.4 ArrayOfFriendlyUrl

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

Collection of **friendly URL** objects in a site.<1>

```
        <xs:complexType name="ArrayOfFriendlyUrl" xmlns:xs="http://www.w3.org/2001/XMLSchema">
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="FriendlyUrl" nillable="true"
    type="tns:FriendlyUrl"/>
          </xs:sequence>
        </xs:complexType>
```

FriendlyUrl: Represents properties of a friendly URL object. A friendly URL is a friendlier version of the underlying SharePoint page URL.

2.2.4.5 FriendlyUrl

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

Represents properties of a **FriendlyUrl** object.<2> **FriendlyUrl** is a friendlier version of the underlying URL of the **page** on the site.

```
        <xs:complexType name="FriendlyUrl" xmlns:xs="http://www.w3.org/2001/XMLSchema">
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="ServerRelativeUrl" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="HostName" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="LastModified" type="xs:dateTime"/>
            <xs:element minOccurs="1" maxOccurs="1" name="LastUpdated" type="xs:dateTime"/>
            <xs:element minOccurs="0" maxOccurs="1" name="Title" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyTitle" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyBrowserTitle"
    type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyDescription"
    type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyKeywords" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="SeoPropertyPriority" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SeoPropertyChangeFrequency"
    type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1"
    name="SeoPropertyInternetSearchEngineInclusion" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="Path" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="SecurityDescriptor"
    type="xs:base64Binary"/>
            <xs:element minOccurs="0" maxOccurs="1" name="CustomProperty" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="TypeOfChange" type="tns:ChangeType"/>
            <xs:element minOccurs="0" maxOccurs="1" name="UniqueId" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
```

ServerRelativeUrl: Server-relative friendly URL for the page on the site .

HostName: Host name for the friendly URL.

LastModified: This value is reserved and MUST be ignored.

LastUpdated: The date and time at which the contents of the page represented by the friendly URL were last updated.

Title: Human-readable title that MAY be displayed when the page is rendered.

SeoPropertyName: Name provided to **web crawlers**.

SeoPropertyTitle: Title provided to the web search crawlers.

SeoPropertyBrowserTitle: Browser title of the page provided to the web search crawlers. This is the text rendered as the title of the web browser when the friendly URL is rendered.

SeoPropertyDescription: Text description of the page provided to the web search crawlers.

SeoPropertyKeywords: Keywords for the page provided to the web search crawlers.

SeoPropertyPriority: Priority for crawling the page provided to the web search crawlers.

SeoPropertyChangeFrequency: Frequency for crawling the page provided to the web search crawlers.

SeoPropertyInternetSearchEngineInclusion: **Boolean** representing whether the page can be crawled provided to the web search crawlers.

Name: Name of the page on the site.

Path: Path of the page on the site that maps to the friendly URL.

SecurityDescriptor: This value is reserved and MUST be ignored.

CustomProperty: This value is reserved and MUST be ignored.

TypeOfChange: An enumeration that identifies the type of change for the URL. MUST be one of the following values.

Value	Meaning
Add	The specified friendly URL was added.
Update	The specified friendly URL was updated.
Delete	The specified friendly URL was deleted.

UniqueId: An identifier string that uniquely identifies the site containing the page.

2.2.4.6 ArrayOfFriendlyUrlBasedWeb

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

A collection of FriendlyUrlBasedWeb objects.<3>

```
    <xs:complexType name="ArrayOfFriendlyUrlBasedWeb"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="FriendlyUrlBasedWeb"
nillable="true" type="tns:FriendlyUrlBasedWeb"/>
      </xs:sequence>
    </xs:complexType>
```

FriendlyUrlBasedWeb: Represents properties of a site that contains friendly URLs.

2.2.4.7 FriendlyUrlBasedWeb

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

Represents properties of a site that contains friendly URLs.<4>

```
    <xs:complexType name="FriendlyUrlBasedWeb" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="SiteId"
xmlns:q3="http://microsoft.com/wsdl/types/" type="q3:guid"/>
        <xs:element minOccurs="1" maxOccurs="1" name="WebId"
xmlns:q4="http://microsoft.com/wsdl/types/" type="q4:guid"/>
        <xs:element minOccurs="0" maxOccurs="1" name="ServerRelativeWebUrl"
type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="TypeOfChange" type="tns:ChangeType"/>
      </xs:sequence>
    </xs:complexType>
```

SiteId: Unique identifier of **site collection** that contains the site that contains friendly URLs.

WebId: Unique identifier of site that contains friendly URLs.

ServerRelativeWebUrl: Server relative URL of site that contains friendly URLs.

TypeOfChange: An enumeration that identifies the type of change for the site. MUST be one of the following values:

Value	Meaning
Add	The site was added.
Update	The site was updated.
Delete	The site was deleted.

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.

2.2.9 Common Data Structures

This specification does not define any common XML schema data structures.

3 Protocol Details

The client side of this protocol is 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. 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 using either HTTP status codes or SOAP faults, as specified previously in this section.

3.1 PublishingServiceSoap Server Details

The following high-level sequence diagram specifies the operation of the protocol. All operations consist of a basic request-response pair, and the protocol server treats each request as an independent transaction that is unrelated to any previous request.

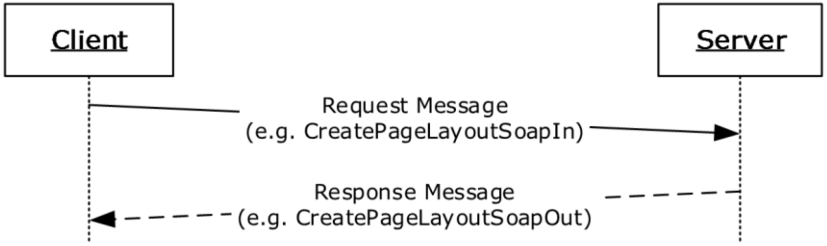

Figure 2: Sequence diagram showing the typical message pattern

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

See [MS-WSSTS] section 2.1 for more information about the underlying abstract data model.

3.1.1.1 Page Layouts

A page layout is a **document** that contains static markup along with dynamic script and controls, including **content placeholders**. A publishing page can be associated with a page layout that serves as the default template for the page. When the page is rendered for display to a user, the content placeholders are filled with the values of specific **fields** from the publishing page. Combining a page layout and a publishing page produces a single HTML page for display. Each page layout has title and description properties along with an associated **content type** that specifies the fields to which the page layout connects. Only a publishing page that has the same content type, or a **descendant content type**, applied can be bound to the page layout. The relationships are specified in the following diagram.

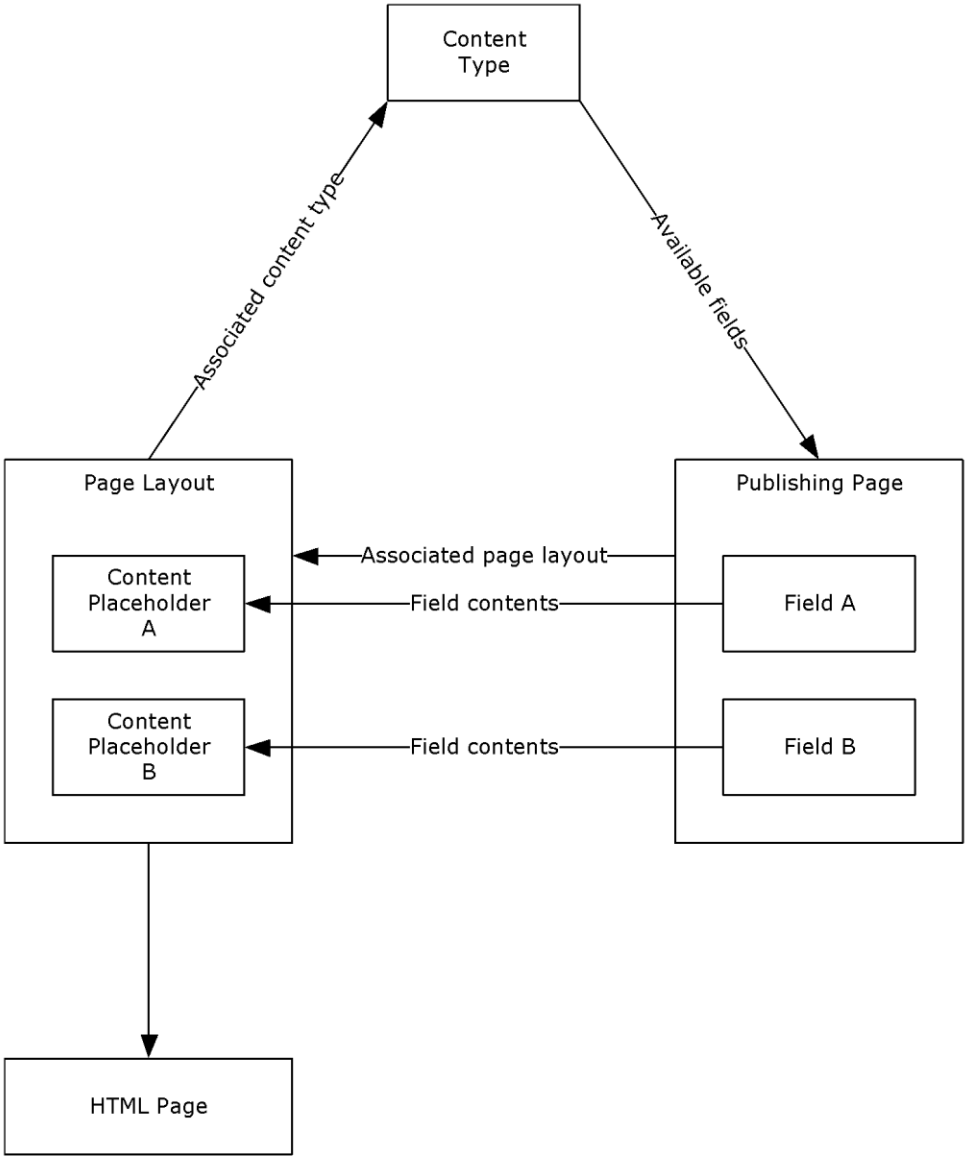

Figure 3: Relationship between page layout, content type, and publishing page

A page layout is often shared by many pages and modifying it will affect all associated pages. A publishing page can be disconnected from its page layout to support the ability to customize the page markup without modifying the page layout. When a publishing page is disconnected from its page layout, the page layout **document stream** is stored in the publishing page so that it can be edited independently of the original page layout. A publishing page can be reconnected to its page layout, which causes the customizations to be discarded. At that point, the publishing page uses the shared page layout to render the page.

3.1.1.2 Scheduling

Scheduling is a term applied to a feature that automates the process of making a **list item** or document available for viewing at a specific time, and removing the list item or document from view at a specific time. The protocol client can use the status operations to retrieve the scheduled start and end times of publishing objects.

3.1.1.3 Variations

Variations is a term applied to a feature that helps manage multilingual content. A **variations** feature automates the process of copying site hierarchies and publishing pages for display in different languages. Editors modify the source variation site and publishing pages it contains, and then the variations feature automatically generates one or more **target variation sites** and publishing pages for one or more languages. The variations feature also keeps track of which publishing page fields are translatable. The protocol client can use the **ExportObjects** and **ImportObjects** operations to extract, and update multilingual content created by such a feature.<5>

3.1.1.4 Friendly URLs

Typical SharePoint page URLs can be long and hard to type or memorize. To solve this problem we have introduced a concept of friendly URLs where a long URL can be shortened to a friendlier version which is typically smaller.<6> The friendly URLs are always mapped to an underlying SharePoint page URLs. The friendly URLs are just a convenient way of accessing these URLs from the browser.

For any transactions with the actual SharePoint object model friendly URLs cannot be used.

3.1.1.5 Change Token Based Containers

The transactions for getting changed friendly URLs involve containers that implement a change token.<7> A change token is an entity that represents the state of the object at that instant. It can be as simple as the current time stamp.

The Content Slice container is a change token based container.

3.1.2 Timers

The protocol server requires a timer to implement the functionality specified by the **Wait** operation in section 3.1.4.9. The Wait timer measures the length of time specified in the operation's input message.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
CreatePageLayout	Creates a new page layout within the specified site collection.
DisconnectPageLayout	Disconnects the specified publishing page from its page layout.
ExportObjects	Exports publishing page content from the specified variant site to a deployment package.<8>
GetAllFriendlyUrls	Obtains all the friendly URLs in the site.
GetChangedFriendlyUrlBasedWebs	Obtains all the sites containing friendly URLs that have changed since the last call.

Operation	Description
GetChangedFriendlyUrls	Obtains all the friendly URLs in the site that have changed since the last call.
GetAllContentSlices	Obtains all the **content slices**.
GetFriendlyUrl	Obtains all the properties associated with a single friendly URL.
GetFriendlyUrlBasedWebs	Obtains all the sites containing friendly URLs.
GetObjectStatus	Retrieves the status of the specified publishing object.
GetObjectStatusCollection	Retrieves the status of a number of publishing objects at the same time.
GetObjectStatusCollectionWithExclusions	Retrieves the status of a range of publishing objects while excluding a specified publishing object.
ImportObjects	Imports translated content from a deployment package into the specified variant site.<9>
ReconnectPageLayout	Reconnects the specified publishing object with its page layout.
Wait	Causes the protocol server to wait for the specified period of time before sending an empty response to the protocol client. <10>

3.1.4.1 CreatePageLayout

This operation is used to create a new page layout within the specified site collection.

```
<wsdl:operation name="CreatePageLayout">
    <wsdl:input message="tns:CreatePageLayoutSoapIn" />
    <wsdl:output message="tns:CreatePageLayoutSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **CreatePageLayoutSoapIn** request message to the protocol server, as specified in section 3.1.4.1.1.1.

The protocol server MUST attempt to create a new page layout with the specified file name within the specified site collection, and associate it with the specified associated content type identifier.

- The protocol server MUST respect the specified file name but MAY<11> override the file name extension if one was specified.
- If title or description values are present in the input, the protocol server MUST set the corresponding page layout properties accordingly.
- If the specified site collection does not exist, the specified page layout name is already in use by a page layout in the site collection, or the specified **content type identifier** does not match the identifier of an existing content type, the protocol server MUST return a SOAP fault.

When the operation finishes execution, the protocol server MUST send a **CreatePageLayoutSoapOut** message that contains a **CreatePageLayoutResult** element set to the absolute URL of the newly created page layout, as specified in section 3.1.4.1.1.2 and section 3.1.4.1.2.2.

3.1.4.1.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
CreatePageLayoutSoapIn	The protocol client sends this message to the protocol server to initiate a **CreatePageLayout** operation.
CreatePageLayoutSoapOut	The protocol server sends this message to the protocol client when the **CreatePageLayout** operation finishes execution.

3.1.4.1.1.1 CreatePageLayoutSoapIn

The protocol client sends this message to the protocol server to initiate a **CreatePageLayout** operation.

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

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

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

3.1.4.1.1.2 CreatePageLayoutSoapOut

The protocol server sends this message to the protocol client when the **CreatePageLayout** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.1.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
CreatePageLayout	The body of the **CreatePageLayoutSoapIn** message, which contains input parameters associated with the **CreatePageLayout** operation.
CreatePageLayoutResponse	The body of the **CreatePageLayoutSoapOut** message, which contains output parameters associated with the **CreatePageLayout** operation.

3.1.4.1.2.1 CreatePageLayout

This element represents the body of the **CreatePageLayoutSoapIn** message and contains input parameters associated with the **CreatePageLayout** operation.

```
<s:element name="CreatePageLayout">
  <s:complexType>
    <s:sequence>
      <s:element name="pageLayoutName">
        <s:simpleType>
          <s:restriction base="s:string">
            <s:minLength value="1"/>
            <s:maxLength value="128"/>
          </s:restriction>
        </s:simpleType>
      </s:element>
      <s:element name="associatedContentTypeId">
        <s:simpleType>
          <s:restriction base="s:string">
            <s:pattern value="0x([0-9A-Fa-f][1-9A-Fa-f]|[1-9A-Fa-f][0-9A-Fa-f]|00[0-9A-Fa-
f]{32})*" />
            <s:minLength value="2"/>
            <s:maxLength value="1026"/>
          </s:restriction>
        </s:simpleType>
      </s:element>
      <s:element name="title" minOccurs="0">
        <s:simpleType>
          <s:restriction base="s:string">
            <s:maxLength value="255"/>
          </s:restriction>
        </s:simpleType>
      </s:element>
      <s:element name="description" type="s:string" minOccurs="0"/>
      <s:element name="siteUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

pageLayoutName: The file name of the new page layout. Additional constraints on the value of this element are specified in [MS-WSSTS] section 2.2.1.

If the file name does not specify an extension, the value MUST be 123 or fewer characters long.

associatedContentTypeId: The content type identifier of the content type to associate with the page layout. Additional constraints on the value of the element are specified in [MS-WSSTS] section 2.1.2.8.1.

title: The title of the new page layout. Additional constraints on the value of this element are specified in [MS-WSSTS] section 2.3.1, Text.

description: A description of the new page layout. Additional constraints on the value of this element are specified in [MS-WSSTS] section 2.3.1, Note.

siteUrl: The absolute URL of the site collection in which to create the page layout.

3.1.4.1.2.2 CreatePageLayoutResponse

This element represents the body of the **CreatePageLayoutSoapOut** message and contains output parameters associated with the **CreatePageLayout** operation.

```
<s:element name="CreatePageLayoutResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="CreatePageLayoutResult" type="s:string"/>
```

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

CreatePageLayoutResult : The absolute URL of the new page layout.

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 DisconnectPageLayout

This operation disconnects the specified publishing page from its page layout.

```
    <wsdl:operation name="DisconnectPageLayout">
        <wsdl:input message="tns:DisconnectPageLayoutSoapIn" />
        <wsdl:output message="tns:DisconnectPageLayoutSoapOut" />
    </wsdl:operation>
```

The protocol client initiates the operation by sending a **DisconnectPageLayoutSoapIn** request message to the protocol server, as specified in section 3.1.4.2.1.1.

The protocol server MUST attempt to disconnect the specified publishing page from its page layout.

If the specified publishing page does not exist, the specified publishing page is not checked out, the specified publishing page is already disconnected from its page layout, or the page layout associated with the publishing page does not exist, the protocol server MUST return a SOAP fault.

When the operation finishes execution, the protocol server MUST send a **DisconnectPageLayoutSoapOut** message, as specified in section 3.1.4.2.1.2.

3.1.4.2.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
DisconnectPageLayoutSoapIn	The protocol client sends this message to the protocol server to initiate a **DisconnectPageLayout** operation.

Message	Description
DisconnectPageLayoutSoapOut	The protocol server sends this message to the protocol client when the **DisconnectPageLayout** operation finishes execution.

3.1.4.2.1.1 DisconnectPageLayoutSoapIn

The protocol client sends this message to the protocol server to initiate a **DisconnectPageLayout** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.2.1.2 DisconnectPageLayoutSoapOut

The protocol server sends this message to the protocol client when the **DisconnectPageLayout** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.2.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
DisconnectPageLayout	The body of the **DisconnectPageLayoutSoapIn** message, which contains input parameters associated with the **DisconnectPageLayout** operation.
DisconnectPageLayoutResponse	The body of the **DisconnectPageLayoutSoapOut** message, which contains output parameters associated with the **DisconnectPageLayout** operation.

3.1.4.2.2.1 DisconnectPageLayout

This element represents the body of the **DisconnectPageLayoutSoapIn** message and contains input parameters associated with the **DisconnectPageLayout** operation.

```
<s:element name="DisconnectPageLayout">
  <s:complexType>
    <s:sequence>
      <s:element name="pageUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
```

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

pageUrl: The absolute URL of the publishing page which is to be disconnected from its page layout.

3.1.4.2.2.2 DisconnectPageLayoutResponse

This element represents the body of the **DisconnectPageLayoutSoapOut** message and contains output parameters associated with the **DisconnectPageLayout** operation.

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

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 ReconnectPageLayout

This operation is used to reconnect the specified publishing page with its page layout.

```
    <wsdl:operation name="ReconnectPageLayout">
        <wsdl:input message="tns:ReconnectPageLayoutSoapIn" />
        <wsdl:output message="tns:ReconnectPageLayoutSoapOut" />
    </wsdl:operation>
```

The protocol client initiates the operation by sending a **ReconnectPageLayoutSoapIn** request message to the protocol server, as specified in section 3.1.4.3.1.1.

The protocol server MUST attempt to reconnect the publishing page to its page layout, as follows:

- If the specified publishing page does not exist, if it is not checked out, or if it is already connected to its associated page layout, the protocol server MUST return a SOAP fault.
- If the associated page layout is no longer available, the publishing page MUST be connected to any one of the available page layouts that have an associated content type matching the content type of the publishing page.
- If no such page layout can be found, the protocol server MUST return a SOAP fault.

When the operation finishes execution, the protocol server MUST send a **ReconnectPageLayoutSoapOut** message containing a **ReconnectPageLayoutResult** element, as specified in section 3.1.4.3.1.2 and section 3.1.4.3.2.2.

See section 3.1.1.1 for more information about the relationship between publishing pages, page layouts, and content types.

3.1.4.3.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
ReconnectPageLayoutSoapIn	The protocol client sends this message to the protocol server to initiate a **ReconnectPageLayout** operation.
ReconnectPageLayoutSoapOut	The protocol server sends this message to the protocol client when the **ReconnectPageLayout** operation finishes execution.

3.1.4.3.1.1 ReconnectPageLayoutSoapIn

The protocol client sends this message to the protocol server to initiate a **ReconnectPageLayout** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.3.1.2 ReconnectPageLayoutSoapOut

The protocol server sends this message to the protocol client when the **ReconnectPageLayout** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.3.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
ReconnectPageLayout	The body of the **ReconnectPageLayoutSoapIn** message, which contains input parameters associated with the **ReconnectPageLayout** operation.
ReconnectPageLayoutResponse	The body of the **ReconnectPageLayoutSoapOut** message, which contains output parameters associated

Element	Description
	with the **ReconnectPageLayout** operation.

3.1.4.3.2.1 ReconnectPageLayout

This element represents the body of the **ReconnectPageLayoutSoapIn** message and contains input parameters associated with the **ReconnectPageLayout** operation.

```
<s:element name="ReconnectPageLayout">
  <s:complexType>
    <s:sequence>
      <s:element name="pageUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

pageUrl: The absolute URL of the publishing page which is to be reconnected to its page layout.

3.1.4.3.2.2 ReconnectPageLayoutResponse

This element represents the body of the **ReconnectPageLayoutSoapOut** message and contains output parameters associated with the **ReconnectPageLayout** operation.

```
<s:element name="ReconnectPageLayoutResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ReconnectPageLayoutResult" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ReconnectPageLayoutResult: Specifies the page layout that the publishing page was connected to. The value MUST be composed of a condition code, followed by a comma, followed by the **site-relative URL** of the page layout.

The condition code MUST be one of the following values:

Value	Meaning
1	The page was reconnected to the same page layout to which it was connected previously.
2	The page was reconnected to a page layout different from the page layout to which it was connected previously; the original page layout was unavailable.

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 ExportObjects

This operation exports translatable publishing page content from a variant site to a deployment package for subsequent translation and importation back into the variant site. See section 3.1.1.3 for details on variations and multilingual content. <12>

```
<wsdl:operation name="ExportObjects">
    <wsdl:input message="tns:ExportObjectsSoapIn" />
    <wsdl:output message="tns:ExportObjectsSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending an **ExportObjectsSoapIn** message to the protocol server, as specified in section 3.1.4.4.1.1.

The protocol server MUST attempt to export translatable publishing pages within the variant site to a deployment package. Required configuration and extension of the nominal deployment package format is specified in section 3.1.4.4.2.2.

If the specified site does not exist, the site is not a variant site, or the operation fails for any other reason, the protocol server MUST either return a SOAP fault or send an **ExportObjectSoapOut** message, as specified in section 3.1.4.4.1.2, with an empty **ExportObjectsResponse** element. The protocol client MUST NOT distinguish between the two cases.

If the operation completes successfully, the protocol server MUST send an **ExportObjectsSoapOut** message to the protocol client. The **ExportObjectsResponse** element, specified in section 3.1.4.4.2.2, MUST contain the binary data of the created deployment package.

3.1.4.4.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
ExportObjectsSoapIn	The protocol client sends this message to the protocol server to initiate an **ExportObjects** operation.
ExportObjectsSoapOut	The protocol server sends this message to the protocol client when the **ExportObjects** operation finishes execution.

3.1.4.4.1.1 ExportObjectsSoapIn

The protocol client sends this message to the protocol server to initiate an **ExportObjects** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.4.1.2 ExportObjectsSoapOut

The protocol server sends this message to the protocol client when the **ExportObjects** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.4.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
ExportObjects	The message body of the **ExportObjectsSoapIn** message, which contains input parameters associated with the **ExportObjects** operation.
ExportObjectsResponse	The body of the **ExportObjectsSoapOut** message. which contains output parameters associated with the **ExportObjects** operation.

3.1.4.4.2.1 ExportObjects

This element represents the message body of the **ExportObjectsSoapIn** message and contains input parameters associated with the **ExportObjects** operation.

```
<s:element name="ExportObjects">
  <s:complexType>
    <s:sequence>
      <s:element name="webUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

webUrl: The absolute URL of the variant site which is to have its content exported.

3.1.4.4.2.2 ExportObjectsResponse

This element represents the body of the **ExportObjectsSoapOut** message and contains output parameters associated with the **ExportObjects** operation.

```
<s:element name="ExportObjectsResponse">
```

```
  <s:complexType>
    <s:sequence>
      <s:element name="ExportObjectsResult" type="s:base64Binary" minOccurs="0"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ExportObjectsResult: This element represents exported publishing page content. The binary value MUST be a valid deployment package as specified in [MS-PRIMEPF]. Required deployment package configuration is as follows:

- The deployment package MUST contain representations of each publishing page in the site.
- The deployment package MUST be compressed. See [MS-PRIMEPF] section 1.3.
- The deployment package MUST contain all publishing page content, not just incremental changes. See [MS-PRIMEPF] section 2.2.4.2.
- The deployment package MUST contain all descendant objects such as content type definitions required by the publishing pages. See [MS-PRIMEPF] section 2.2.4.3.
- The deployment package MUST NOT contain any security attributes. See [MS-PRIMEPF] section 2.2.4.4.
- The deployment package MUST contain the current version of each publishing page in the site as specified in [MS-PRIMEPF] section 2.2.4.5.
- The deployment package MUST contain an additional **XML** file named VariationsLanguageSettings.xml containing content that MUST adhere to the definition of the **LanguageSettings** element as specified in section 3.1.4.4.2.3.
- If the content contains any translatable fields, the package MUST contain an additional XML file named TranslatableFieldSettings.xml containing content that MUST adhere to the definition of the **Fields** element as specified in section 3.1.4.4.2.4.

3.1.4.4.2.3 Language Settings

This element represents the body of the VariationsLanguageSettings.xml document and MUST adhere to the following **XML schema definition (XSD)**.

```
<s:element name="LanguageSettings">
  <s:complexType>
    <s:sequence>
      <s:element name="SourceLanguage" type="s:string"/>
      <s:element name="TargetLanguage" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

SourceLanguage: The language of the source variation site. The value MUST be a valid language tag as specified in [RFC1766] section 2.

TargetLanguage: The language of the target variation site, and therefore the desired language of the content. The value MUST be a valid language tag as specified in [RFC1766] section 2.

3.1.4.4.2.4 Fields

This element represents the body of the TranslatableFieldSettings.xml document and MUST adhere to the following XSD.

```
<s:element name="Fields">
  <s:complexType>
    <s:sequence>
      <s:element name="Field" maxOccurs="unbounded">
        <s:complexType>
          <s:simpleContent>
            <s:extension base="s:string">              <s:attribute name="Id" use="required">
<xs:simpleType>                  <xs:restriction base="xs:string">
<xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{12}"/>                  </xs:restriction>                 </xs:simpleType>
</s:attribute>
            </s:extension>
          </s:simpleContent>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

Field: Each element represents a field which contains a string value to translate. The element value represents the display name of the field.

Field.Id: The **GUID** of the field represented by the associated **Field** element.

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 ImportObjects

This operation is used to import translated content back into the variant site collection from which it was exported.<13>

```
<wsdl:operation name="ImportObjects">
    <wsdl:input message="tns:ImportObjectsSoapIn" />
    <wsdl:output message="tns:ImportObjectsSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending an **ImportObjectsSoapIn** message to the protocol server, as specified in section 3.1.4.5.1.1.

The protocol server MUST attempt to import the specified content into the specified site collection. If an imported **list item** or **file** collides with an object that already exists in the store, any existing checked-out versions of the item MUST be checked in and the imported object MUST be stored as a new minor version of the existing document.

If the specified site collection does not exist, the format of the content is not valid, or the **ImportObjects** operation fails for any other reason, the protocol server MUST either return a SOAP fault or send an **ImportObjectsSoapOut** message, as specified in section 3.1.4.5.1.2, with the **ImportObjectsResult** element set to False. The protocol client MUST NOT distinguish between the two cases.

If the operation completes successfully, the protocol server MUST send an **ImportObjectsSoapOut** message to the protocol server. The **ImportObjectsResult** element, specified in section 3.1.4.5.2.2, MUST be set to **true**.

3.1.4.5.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
ImportObjectsSoapIn	The protocol client sends this message to the protocol server to initiate an **ImportObjects** operation.
ImportObjectsSoapOut	The protocol server sends this message to the protocol client when the **ImportObjects** operation finishes execution.

3.1.4.5.1.1 ImportObjectsSoapIn

The protocol client sends this message to the protocol server to initiate an **ImportObjects** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.5.1.2 ImportObjectsSoapOut

The protocol server sends this message to the protocol client when the **ImportObjects** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.5.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
ImportObjects	The message body of the **ImportObjectsSoapIn**

Element	Description
	message, which contains input parameters associated with the **ImportObjects** operation.
ImportObjectsResponse	The body of the **ImportObjectsSoapOut** message, which contains output parameters associated with the **ImportObjects** operation.

3.1.4.5.2.1 ImportObjects

This element represents the message body of the **ImportObjectsSoapIn** message and contains input parameters associated with the **ImportObjects** operation.

```
<s:element name="ImportObjects">
  <s:complexType>
    <s:sequence>
      <s:element name="siteUrl" type="s:string"/>
      <s:element name="fileContent" type="s:base64Binary"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

siteUrl: Represents the absolute URL of the site collection into which to import the content.

fileContent: Represents the packaged content to import. The binary value MUST be a valid deployment package that was created using the same deployment settings as specified for the **ExportObjectsResponse** element in section 3.1.4.4.2.2.

3.1.4.5.2.2 ImportObjectsResponse

This element represents the body of the **ImportObjectsSoapOut** message and contains output parameters associated with the **ImportObjects** operation.

```
<s:element name="ImportObjectsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ImportObjectsResult" type="s:boolean"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

ImportObjectsResult: This element indicates whether the **ImportObjects** operation was successful.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 GetObjectStatus

This operation obtains status information for the specified publishing object.

```
<wsdl:operation name="GetObjectStatus">
    <wsdl:input message="tns:GetObjectStatusSoapIn" />
    <wsdl:output message="tns:GetObjectStatusSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetObjectStatusSoapIn** message to the protocol server, as specified in section 3.1.4.6.1.1.

The protocol server MUST attempt to build a **PublishingObjectStatus** element, as specified in section 2.2.4.1, to send to the protocol client. The following conditions must be filled:

- If the specified URL uses either the javascript://, mailto:// or news:// scheme, the **ObjectType** element MUST be set to "Undefined".
- If the specified URL does not refer to an object within a site collection on the protocol server, or if the protocol client is unauthorized to access it, the **ObjectType** element MUST be set to either "FileNotFound" or "AccessDenied". The protocol client MUST NOT distinguish between the two cases.
- If the specified object is found but it is not a list item, document, or file, the **ObjectType** element MUST be set to "Undefined".
- If the specified publishing object is found and all properties are retrieved successfully, the **ObjectType** element MUST be set to "File".
- If any other failures are encountered while processing the operation, the protocol server MUST return a SOAP fault.

When the operation finishes execution, the protocol server MUST send a **GetObjectStatusSoapOut** message that includes the **PublishingObjectStatus** element, as specified in section 3.1.4.6.1.2 and section 3.1.4.6.2.2.

3.1.4.6.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetObjectStatusSoapIn	The protocol client sends this message to the protocol server to initiate a **GetObjectStatus** operation.
GetObjectStatusSoapOut	The protocol server sends this message to the protocol client when the **GetObjectStatus** operation finishes execution.

3.1.4.6.1.1 GetObjectStatusSoapIn

The protocol client sends this message to the protocol server to initiate a **GetObjectStatus** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.6.1.2 GetObjectStatusSoapOut

The protocol server sends this message to the protocol client when the **GetObjectStatus** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.6.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetObjectStatus	The body of the **GetObjectStatusSoapIn** message, which contains input parameters associated with the **GetObjectStatus** operation.
GetObjectStatusResponse	The body of the **GetObjectStatusSoapOut** message, which contains output parameters associated with the **GetObjectStatus** operation.

3.1.4.6.2.1 GetObjectStatus

This element represents the body of the **GetObjectStatusSoapIn** message and contains input parameters associated with the **GetObjectStatus** operation.

```
<s:element name="GetObjectStatus">
  <s:complexType>
    <s:sequence>
      <s:element name="objectUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

objectUrl: The absolute URL of the publishing object for which status is to be fetched.

3.1.4.6.2.2 GetObjectStatusResponse

This element represents the body of the **GetObjectStatusSoapOut** message and contains output parameters associated with the **GetObjectStatus** operation.

```
    <s:element name="GetObjectStatusResponse">
      <s:complexType>
        <s:sequence>
          <s:element name="GetObjectStatusResult" type="tns:PublishingObjectStatus"/>
        </s:sequence>
      </s:complexType>
    </s:element>
```

GetObjectStatusResult: Contains publishing object properties specified in section 2.2.4.1, **PublishingObjectStatus**.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 GetObjectStatusCollection

This operation is intended to obtain status information for a collection of specified publishing objects, performing the **GetObjectStatus** operation in bulk. The operation is deprecated because it is of no practical use as specified and there are no existing protocol clients that depend on it. A protocol server is still considered compliant if it does not support this operation.

```
    <wsdl:operation name="GetObjectStatusCollection">
        <wsdl:input message="tns:GetObjectStatusCollectionSoapIn" />
        <wsdl:output message="tns:GetObjectStatusCollectionSoapOut" />
    </wsdl:operation>
```

The protocol client initiates the operation by sending a **GetObjectStatusCollectionSoapIn** message to the protocol server as specified in section 3.1.4.7.1.1.

Regardless of the input, the protocol server MUST return a SOAP fault.

3.1.4.7.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetObjectStatusCollectionSoapIn	The protocol client sends this message to the protocol server to initiate a **GetObjectStatusCollection**

Message	Description
	operation.
GetObjectStatusCollectionSoapOut	The protocol server sends this message to the protocol client when the **GetObjectStatusCollection** operation finishes execution.

3.1.4.7.1.1 GetObjectStatusCollectionSoapIn

The protocol client sends this message to the protocol server to initiate a **GetObjectStatusCollection** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.7.1.2 GetObjectStatusCollectionSoapOut

The protocol server sends this message to the protocol client when the **GetObjectStatusCollection** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.7.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetObjectStatusCollection	The body of the **GetObjectStatusCollectionSoapIn** message, which contains input parameters associated with the **GetObjectStatusCollection** operation.
GetObjectStatusCollectionResponse	The body of the **GetObjectStatusCollectionSoapOut** message, which contains output parameters associated with the **GetObjectStatusCollection** operation.

3.1.4.7.2.1 GetObjectStatusCollection

This element represents the body of the **GetObjectStatusCollectionSoapIn** message, and contains input parameters associated with the **GetObjectStatusCollection** operation.

```
<s:element name="GetObjectStatusCollection">

  <s:complexType>
```

```
    <s:sequence>
      <s:element name="objectUrls" type="tns:ArrayOfString"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

objectUrls: Each element in the collection specifies the absolute URL of a publishing object for which status is to be retrieved.

3.1.4.7.2.2 GetObjectStatusCollectionResponse

This element represents the body of the **GetObjectStatusCollectionSoapOut** message and contains output parameters associated with the **GetObjectStatusCollection** operation.

```
<s:element name="GetObjectStatusCollectionResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetObjectStatusCollectionResult" 
type="tns:ArrayOfPublishingObjectStatus"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetObjectStatusCollectionResult: Each element in the collection contains content that represents publishing object properties as specified section 2.2.4.3, **ArrayOfPublishingObjectStatus**.

3.1.4.7.3 Complex Types

None.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 GetObjectStatusCollectionWithExclusions

This operation obtains status information for a collection of specified publishing objects, excluding a specified publishing object.

```
<wsdl:operation name="GetObjectStatusCollectionWithExclusions">
    <wsdl:input message="tns:GetObjectStatusCollectionWithExclusionsSoapIn" />
    <wsdl:output message="tns:GetObjectStatusCollectionWithExclusionsSoapOut" />
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetObjectStatusCollectionWithExclusionsSoapIn** message to the protocol server, formatted as specified in section 3.1.4.8.1.1.

If the **thisPageUrl** element is set to a URL with a domain not equivalent to the domain of the protocol server, the protocol server MUST return a SOAP fault. Otherwise, the protocol server MUST perform the **GetObjectStatus** operation on each URL in the specified collection, as specified in section 3.1.4.6, with the following modifications:

- If the **thisPageUrl** element has the same value as the current URL from the input collection, as in the URL that identifies the object for which status is to be retrieved, the protocol server MUST set the corresponding **ObjectType** element to Undefined.
- In the case of an unexpected failure, rather than returning a SOAP fault the protocol server MUST set the corresponding **ObjectType** element to "UnrecoverableFailure" and set the **Description** element to a brief failure notification message.

The resulting **PublishingObjectStatus** elements MUST be added to an **ArrayOfPublishingObjectStatus** element in the same order that the corresponding input URLs were found. When the operation finishes execution, the protocol server MUST send a **GetObjectStatusCollectionWithExclusionsSoapOut** message containing the result values as specified in section 3.1.4.8.1.2 and section 3.1.4.8.2.2.

3.1.4.8.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetObjectStatusCollectionWithExclusionsSoapIn	The protocol client sends this message to the protocol server to initiate a **GetObjectStatusCollectionWithExclusions** operation.
GetObjectStatusCollectionWithExclusionsSoapOut	The protocol server sends this message to the protocol client when the **GetObjectStatusCollectionWithExclusions** operation finishes execution.

3.1.4.8.1.1 GetObjectStatusCollectionWithExclusionsSoapIn

The protocol client sends this message to the protocol server to initiate a **GetObjectStatusCollectionWithExclusions** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.8.1.2 GetObjectStatusCollectionWithExclusionsSoapOut

The protocol server sends this message to the protocol client when the **GetObjectStatusCollectionWithExclusions** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.8.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetObjectStatusCollectionWithExclusions	The body of the **GetObjectStatusCollectionWithExclusionsSoapIn** message, which contains input parameters associated with the **GetObjectStatusCollectionWithExclusions** operation.
GetObjectStatusCollectionWithExclusionsResponse	The body of the **GetObjectStatusCollectionWithExclusionsSoapOut** message, which contains output parameters associated with the **GetObjectStatusCollectionWithExclusions** operation.

3.1.4.8.2.1 GetObjectStatusCollectionWithExclusions

This element represents the body of the **GetObjectStatusCollectionWithExclusionsSoapIn** message and contains input parameters associated with the **GetObjectStatusCollectionWithExclusions** operation.

```
<s:element name="GetObjectStatusCollectionWithExclusions">
  <s:complexType>
    <s:sequence>
      <s:element name="objectUrls" type="tns:ArrayOfString"/>
      <s:element name="thisPageUrl" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

objectUrls: Each element in the collection specifies the absolute URL of a publishing object for which status is retrieved.

thisPageUrl: The absolute URL of a publishing object that has properties which are not retrieved even if the URL is found in the **objectUrls** collection. The domain specified by this URL MUST match the domain of the protocol server.

3.1.4.8.2.2 GetObjectStatusCollectionWithExclusionsResponse

This element represents the body of the **GetObjectStatusCollectionWithExclusionsSoapOut** message and contains output parameters associated with the **GetObjectStatusCollectionWithExclusions** operation.

```
<s:element name="GetObjectStatusCollectionWithExclusionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetObjectStatusCollectionWithExclusionsResult"
type="tns:ArrayOfPublishingObjectStatus"/>
```

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

GetObjectStatusCollectionWithExclusionsResult: Each element in the collection contains content that represents various publishing object properties as specified in section 2.2.4.3, **ArrayOfPublishingObjectStatus**, subject to the restrictions specified in section 3.1.4.8.

3.1.4.8.3 Complex Types

None.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 Wait

This operation forces the protocol server to wait for the specified amount of time before sending a response to the protocol client.<14>

```
    <wsdl:operation name="Wait">
        <wsdl:input message="tns:WaitSoapIn" />
        <wsdl:output message="tns:WaitSoapOut" />
    </wsdl:operation>
```

The protocol client initiates the operation by sending a **WaitSoapIn** message to the protocol server, as specified in section 3.1.4.9.1.1.

The protocol server MUST initiate a millisecond timer that expires after the period designated by the **millisecondsToWait** element. When the timer expires, the protocol server MUST send a **WaitSoapOut** message as specified in section 3.1.4.9.1.2. If the operation fails for any reason, the protocol server MUST return a SOAP fault.

3.1.4.9.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
WaitSoapIn	The protocol client sends this message to the protocol server to initiate a **Wait** operation.

Message	Description
WaitSoapOut	The protocol server sends this message to the protocol client when the **Wait** operation finishes execution.

3.1.4.9.1.1 WaitSoapIn

The protocol client sends this message to the protocol server to initiate a **Wait** operation.

The SOAP action value of the message is specified as:

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

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

3.1.4.9.1.2 WaitSoapOut

The protocol server sends this message to the protocol client when the **Wait** operation finishes execution.

The SOAP action value of the message is specified as:

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

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

3.1.4.9.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
Wait	The body of the **WaitSoapIn** message, which contains input parameters associated the **Wait** operation.
WaitResponse	The body of the **WaitSoapOut** message, which contains output parameters associated with the **Wait** operation.

3.1.4.9.2.1 Wait

This element represents the body of the **WaitSoapIn** message and contains input parameters associated the **Wait** operation.

```
<s:element name="Wait">
  <s:complexType>
    <s:sequence>
      <s:element name="millisecondsToWait" type="s:int"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

millisecondsToWait: The number of milliseconds to wait before responding. The value MUST be non-negative.

3.1.4.9.2.2 WaitResponse

This element represents the body of the **WaitSoapOut** message and contains output parameters associated with the **Wait** operation.

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

3.1.4.9.3 Complex Types

None.

3.1.4.9.4 Simple Types

None.

3.1.4.9.5 Attributes

None.

3.1.4.9.6 Groups

None.

3.1.4.9.7 Attribute Groups

None.

3.1.4.10 GetAllContentSlices

This operation obtains all the content slices.<15>

```
<wsdl:operation name="GetAllContentSlices" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetAllContentSlicesSoapIn"/>
  <wsdl:output message="tns:GetAllContentSlicesSoapOut"/>
</wsdl:operation>
```

A content slice is a container of sites (2) that contain friendly URLs.

The protocol client initiates the operation by sending a **GetAllContentSlicesSoapIn** message to the protocol server, as specified in section 3.1.4.10.1.1.

The protocol server MUST return **ArrayOfContentSlice** element, as specified in the section 3.1.4.10.3.1, to send to the protocol client. The following conditions must be filled:

- If there is 1 or more content slices then the service MUST return properties related all the content slices.
- The content slices MUST be change token based containers.
- In case of error the server MUST return a SOAP fault.

3.1.4.10.1 Messages

The following table summarizes the set of **WSDL message** definitions that are specific to this operation.

Message	Description
GetAllContentSlicesSoapIn	The request WSDL message for the **GetAllContentSlices** WSDL operation.
GetAllContentSlicesSoapOut	The response WSDL message for the **GetAllContentSlices** WSDL operation.

3.1.4.10.1.1 GetAllContentSlicesSoapIn

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

The SOAP action value is:

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

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

3.1.4.10.1.2 GetAllContentSlicesSoapOut

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

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

3.1.4.10.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetAllContentSlices	The input data for the **GetAllContentSlices** WSDL operation.
GetAllContentSlicesResponse	The result data for the **GetAllContentSlices** WSDL operation.

3.1.4.10.2.1 GetAllConentSlices

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

```
<xs:element name="GetAllContentSlices" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType/>
</xs:element>
```

3.1.4.10.2.2 GetAllContentSlicesResponse

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

This element represents the body of the **GetAllContentSlicesSoapOut** message and contains output parameters associated with the **GetAllContentSlices** operation.

```
<xs:element name="GetAllContentSlicesResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
```

```
      <xs:element minOccurs="0" maxOccurs="1" name="GetAllContentSlicesResult" 
type="tns:ArrayOfContentSlice"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetAllContentSlicesResult Contains content slice properties specified in section 3.1.4.10.3.1.

3.1.4.10.3 Complex Types

The following table summarizes the XML schema complex type definitions that are specific to this operation.

Complex type	Description
ArrayOfContentSlice	Collection of properties of all content slices.
ContentSlice	Object representing properties of the content slice.

3.1.4.10.3.1 ArrayOfContentSlice

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

```
<xs:complexType name="ArrayOfContentSlice" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ContentSlice" nillable="true" 
type="tns:ContentSlice"/>
  </xs:sequence>
</xs:complexType>
```

ContentSlice: Object representing properties of the content slice.

3.1.4.10.3.2 ContentSlice

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

```
<xs:complexType name="ContentSlice" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Id" 
xmlns:q1="http://microsoft.com/wsdl/types/" type="q1:guid"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ChangeToken" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Id: Unique identifier of the content slice.

ChangeToken: Opaque token representing the state of the content slice at the given instant.

3.1.4.10.4 Simple Types

None.

3.1.4.10.5 Attributes

None.

3.1.4.10.6 Groups

None.

3.1.4.10.7 Attribute Groups

None.

3.1.4.11 GetFriendlyUrlBasedWebs

This operation obtains all the sites containing friendly URLs.<16>

```
<wsdl:operation name="GetFriendlyUrlBasedWebs" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetFriendlyUrlBasedWebsSoapIn"/>
  <wsdl:output message="tns:GetFriendlyUrlBasedWebsSoapOut"/>
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetFriendlyUrlBasedWebsSoapIn** message to the protocol server, as specified in section 3.1.4.11.1.1
The protocol server MUST attempt to build a **ArrayOfFriendlyUrlBasedWeb** element, as specified in section 2.2.4.6, to send to the protocol client. The following conditions must be filled:

- The protocol server MUST enumerate all the sites containing friendly URLs and MUST create a **FriendlyUrlBasedWeb** object for each site and add it to the **ArrayOfFriendlyUrlBasedWeb** which is returned.
- If there are a large number sites then the protocol server MUST return **ArrayOfFriendlyUrlBasedWeb** for a subset of those sites along with a pagingToken.
- The protocol server MUST return different subset of the **ArrayOfFriendlyUrlBasedWeb** for different values of pagingToken.
- Protocol server MUST return a SOAP fault if there is an error processing the pagingToken.
- If the contentPartitionId specified does not exist or is null the protocol server MUST return a SOAP fault.

3.1.4.11.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetFriendlyUrlBasedWebsSoapIn	The request WSDL message for the **GetFriendlyUrlBasedWebs** WSDL operation.
GetFriendlyUrlBasedWebsSoapOut	The response WSDL message for the **GetFriendlyUrlBasedWebs** WSDL operation.

3.1.4.11.1.1 GetFriendlyUrlBasedWebsSoapIn

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

The SOAP action value is:

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

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

3.1.4.11.1.2 GetFriendlyUrlBasedWebSoapOut

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

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

3.1.4.11.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetFriendlyUrlBasedWebs	The input data for the **GetFriendlyUrlBasedWebs** WSDL operation.
GetFriendlyUrlBasedWebsResponse	The result data for the **GetFriendlyUrlBasedWebs** WSDL operation.

3.1.4.11.2.1 GetFriendlyUrlBasedWebs

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

```
<xs:element name="GetFriendlyUrlBasedWebs" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="contentPartitionId" 
xmlns:q2="http://microsoft.com/wsdl/types/" type="q2:guid"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

contentPartitionId: PartitionId of the content slice.

pagingToken: An opaque token passed by the protocol client which MUST be used to obtain a different subset of sites. If null this parameter MUST be ignored by the protocol server.

3.1.4.11.2.2 GetFriendlyUrlBasedWebsResponse

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

```
<xs:element name="GetFriendlyUrlBasedWebsResponse" 
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetFriendlyUrlBasedWebsResult" 
type="tns:ArrayOfFriendlyUrlBasedWeb"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetFriendlyUrlBasedWebsResult: Contains array of properties for sites containing friendly URLs specified in section 2.2.4.6

pagingToken: An opaque token returned to the protocol client which MUST be used to obtain a different subset of sites when there is a large number of sites to be returned by the protocol server. This MUST be set to null by the protocol server when there are no more sites to return.

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 GetChangedFriendlyUrlBasedWebs

This operation obtains all the sites containing friendly URLs that have changed since the last call.<17>

```
<wsdl:operation name="GetChangedFriendlyUrlBasedWebs"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetChangedFriendlyUrlBasedWebsSoapIn"/>
  <wsdl:output message="tns:GetChangedFriendlyUrlBasedWebsSoapOut"/>
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetChangedFriendlyUrlBasedWebs** message to the protocol server, as specified in section 3.1.4.12.2.1

The protocol server MUST build **ArrayOfFriendlyUrlBasedWeb** element, as specified in section 2.2.4.6, to send to the protocol client. The following conditions must be filled:

The protocol server MUST enumerate all the sites containing friendly URLs and that have changed since given changeToken and MUST create a **FriendlyUrlBasedWeb** object for each site and add it to the **ArrayOfFriendlyUrlBasedWeb** which is returned.

- If there are a large number sites that have changed then the protocol server MUST return **ArrayOfFriendlyUrlBasedWeb** for a subset of those sites (2) along with a pagingToken.
- The protocol server MUST return different subset of the **ArrayOfFriendlyUrlBasedWeb** for different values of pagingToken.
- Protocol server MUST return a SOAP fault if there is an error processing the pagingToken or changeToken.
- If the contentPartitionId specified does not exist or is null the protocol server MUST return a SOAP fault.

3.1.4.12.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetChangedFriendlyUrlBasedWebsSoapIn	The request WSDL message for the **GetChangedFriendlyUrlBasedWebs** WSDL operation.
GetChangedFriendlyUrlBasedWebsSoapOut	The response WSDL message for the **GetChangedFriendlyUrlBasedWebs** WSDL operation.

3.1.4.12.1.1 GetChangedFriendlyUrlBasedWebsSoapIn

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

The SOAP action value is:

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

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

3.1.4.12.1.2 GetChangedFriendlyUrlBasedWebsSoapOut

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

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

3.1.4.12.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetChangedFriendlyUrlBasedWebs	The input data for the **GetChangedFriendlyUrlBasedWebs** WSDL operation.
GetChangedFriendlyUrlBasedWebsResponse	The result data for the **GetChangedFriendlyUrlBasedWebs** WSDL operation.

3.1.4.12.2.1 GetChangedFriendlyUrlBasedWebs

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

```
<xs:element name="GetChangedFriendlyUrlBasedWebs"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="contentPartitionId"
xmlns:q5="http://microsoft.com/wsdl/types/" type="q5:guid"/>
      <xs:element minOccurs="0" maxOccurs="1" name="changeToken" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

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

contentPartitionId: Id of the content slice container for getting changed sites.

changeToken: Opaque token representing the state of the content slice at the given instant.

pagingToken: An opaque token passed by the protocol client which MUST be used to obtain a different subset of sites. If null this parameter MUST be ignored by the protocol server.

3.1.4.12.2.2 GetChangedFriendlyUrlBasedWebsResponse

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

```
<xs:element name="GetChangedFriendlyUrlBasedWebsResponse" 
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetChangedFriendlyUrlBasedWebsResult" 
type="tns:ArrayOfFriendlyUrlBasedWeb"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetChangedFriendlyUrlBasedWebsResult: Contains array of site properties specified in section 2.2.4.6.

pagingToken: An opaque token returned to the protocol client which MUST be used to obtain a different subset of sites when there is a large number of sites to be returned by the protocol server. This MUST be set to null by the protocol server when there are no more sites to return.

3.1.4.12.3 Complex Types

None.

3.1.4.12.4 Simple Types

None.

3.1.4.12.5 Attributes

None.

3.1.4.12.6 Groups

None.

3.1.4.12.7 Attribute Groups

None.

3.1.4.13 GetAllFriendlyUrls

This operation obtains all the friendly URLs in the site.<18>

```
<wsdl:operation name="GetAllFriendlyUrls" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetAllFriendlyUrlsSoapIn"/>
```

```
      <wsdl:output message="tns:GetAllFriendlyUrlsSoapOut"/>
    </wsdl:operation>
```

The protocol client initiates the operation by sending a **GetAllFriendlyUrlsSoapIn** message to the protocol server, as specified in section 3.1.4.13.1.1

The protocol server MUST attempt to build an **ArrayOfFriendlyUrl** element, as specified in section 3.1.4.13.1.2 and section 2.2.4.4, to send to the protocol client. The following conditions must be filled:

- The protocol server MUST enumerate all the friendly URLs in the site and MUST create a **FriendlyUrl** object, as specified in section 2.2.4.5, for each friendly URL and add it to the ArrayOfFriendlyUrl which is returned.
- If there are a large number of friendly URLs then the protocol server MUST return ArrayOfFriendlyUrl for a subset of those URL along with a **pagingToken**, as specified in section 3.1.4.13.2.2 under **GetAllFriendlyUrlsResponse**.
- The protocol server MUST return different subset of the ArrayOfFriendlyUrl for different values of pagingToken.
- Protocol server MUST return a SOAP fault if there is an error processing the pagingToken.
- If the **siteId** (section 3.1.4.13.2.1) or **webId** (section 3.1.4.13.2.1) specified does not exist or is null the protocol server MUST return a SOAP fault.

When the operation finishes execution, the protocol server MUST send a **GetAllFriendlyUrlsSoapOut** message (section 3.1.4.14.1.2) that includes the ArrayOfFriendlyUrl element.

3.1.4.13.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetAllFriendlyUrlsSoapIn	The request WSDL message for the **GetAllFriendlyUrls** WSDL operation.
GetAllFriendlyUrlsSoapOut	The response WSDL message for the **GetAllFriendlyUrls** WSDL operation.

3.1.4.13.1.1 GetAllFriendlyUrlsSoapIn

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

The SOAP action value is:

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

3.1.4.13.1.2 GetAllFriendlyUrlsSoapOut

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

The SOAP body contains the **GetAllFriendlyUrlsResponse** element

3.1.4.13.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetAllFriendlyUrls	The input data for the **GetAllFriendlyUrls** WSDL operation.
GetAllFriendlyUrlsResponse	The result data for the **GetAllFriendlyUrls** WSDL operation.

3.1.4.13.2.1 GetAllFriendlyUrls

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

```
<xs:element name="GetAllFriendlyUrls" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="siteId"
xmlns:q6="http://microsoft.com/wsdl/types/" type="q6:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="webId"
xmlns:q7="http://microsoft.com/wsdl/types/" type="q7:guid"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

siteId: Unique identifier of the site collection that contains the site that contains friendly URLs.

webId: Unique identifier of the site that contains friendly URLs.

pagingToken: An opaque token passed by the protocol client which MUST be used to obtain a different subset of friendly URLs. If null this parameter MUST be ignored by the protocol server.

3.1.4.13.2.2 GetAllFriendlyUrlsResponse

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

```
<xs:element name="GetAllFriendlyUrlsResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetAllFriendlyUrlsResult"
type="tns:ArrayOfFriendlyUrl"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetAllFriendlyUrlsResult Contains array of friendly URL properties specified in section 2.2.4.4

pagingToken: An opaque token returned to the protocol client which MUST be used to obtain a different subset of friendly URLs when there is a large number of URLs to be returned by the protocol server. This MUST be set to null by the protocol server when there are no more friendly URLs to return.

3.1.4.13.3 Complex Types

None.

3.1.4.13.4 Simple Types

None.

3.1.4.13.5 Attributes

None.

3.1.4.13.6 Groups

None.

3.1.4.13.7 Attribute Groups

None.

3.1.4.14 GetChangedFriendlyUrls

This operation obtains all the friendly URLs in the site that have changed since the last call.<19>

```
<wsdl:operation name="GetChangedFriendlyUrls" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetChangedFriendlyUrlsSoapIn"/>
  <wsdl:output message="tns:GetChangedFriendlyUrlsSoapOut"/>
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetChangedFriendlyUrls** message to the protocol server, as specified in section 3.1.4.14.2.1
The protocol server MUST attempt to build a **ArrayOfFriendlyUrl** element, as specified in section 2.2.4.4, to send to the protocol client. The following conditions must be filled:

- The protocol server MUST enumerate all the friendly URLs and have changed and MUST create a **FriendlyUrl** object for each friendly URL and add it to the **ArrayOfFriendlyUrl** which is returned.
- If there are a large number friendly URLs that have changed then the protocol server MUST return **ArrayOfFriendlyUrl** for a subset of those friendly URLs along with a pagingToken.
- The protocol server MUST return different subset of the **ArrayOfFriendlyUrl** for different values of pagingToken.
- Protocol server MUST return a SOAP fault if there is an error processing the pagingToken.
- If the siteId or webId specified does not exist or is null the protocol server MUST return a SOAP fault.

3.1.4.14.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetChangedFriendlyUrlsSoapIn	The request WSDL message for the **GetChangedFriendlyUrls** WSDL operation.
GetChangedFriendlyUrlsSoapOut	The response WSDL message for the **GetChangedFriendlyUrls** WSDL operation.

3.1.4.14.1.1 GetChangedFriendlyUrlsSoapIn

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

The SOAP action value is:

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

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

3.1.4.14.1.2 GetChangedFriendlyUrlsSoapOut

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

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

3.1.4.14.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetChangedFriendlyUrls	The input data for the **GetChangedFriendlyUrls** WSDL operation.
GetChangedFriendlyUrlsResponse	The result data for the **GetChangedFriendlyUrls** WSDL operation.

3.1.4.14.2.1 GetChangedFriendlyUrls

The input data for the GetChangedFriendlyUrls WSDL operation.

```
<xs:element name="GetChangedFriendlyUrls" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="siteId" 
xmlns:q8="http://microsoft.com/wsdl/types/" type="q8:guid"/>
      <xs:element minOccurs="1" maxOccurs="1" name="webId" 
xmlns:q9="http://microsoft.com/wsdl/types/" type="q9:guid"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

pagingToken: An opaque token passed by the protocol client which MUST be used to obtain a different subset of friendly URLs. If null this parameter MUST be ignored by the protocol server.

siteId: Unique identifier of site collection that contains the site that contains friendly URLs.

webId: Unique identifier of site that contains friendly URLs.

3.1.4.14.2.2 GetChangedFriendlyUrlsResponse

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

```
<xs:element name="GetChangedFriendlyUrlsResponse" 
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
```

```
      <xs:element minOccurs="0" maxOccurs="1" name="GetChangedFriendlyUrlsResult" 
type="tns:ArrayOfFriendlyUrl"/>
      <xs:element minOccurs="0" maxOccurs="1" name="pagingToken" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetChangedFriendlyUrlsResult: Contains array of friendly URL properties specified in section 2.2.4.4.

pagingToken: An opaque token returned to the protocol client which MUST be used to obtain a different subset of friendly URLs when there is a large number of URLs to be returned by the protocol server. This MUST be set to null by the protocol server when there are no more friendly URLs to return.

3.1.4.14.3 Complex Types

None.

3.1.4.14.4 Simple Types

None.

3.1.4.14.5 Attributes

None.

3.1.4.14.6 Groups

None.

3.1.4.14.7 Attribute Groups

None.

3.1.4.15 GetFriendlyUrl

This operation obtains all the properties associated with a single friendly URL.<20>

```
<wsdl:operation name="GetFriendlyUrl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetFriendlyUrlSoapIn"/>
  <wsdl:output message="tns:GetFriendlyUrlSoapOut"/>
</wsdl:operation>
```

The protocol client initiates the operation by sending a **GetFriendlyUrlSoapIn** message to the protocol server, as specified in section 3.1.4.15.1.1
The protocol server MUST attempt to build a **FriendlyUrl** element, as specified in section 2.2.4.5, to send to the protocol client. The following conditions must be filled:

- The protocol server MUST create a **FriendlyUrl** object for friendly URL identified by the specified input URL.
- If there is an error creating the **FriendlyUrl** object protocol server MUST return a SOAP fault.

3.1.4.15.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.

Message	Description
GetFriendlyUrlSoapIn	The request WSDL message for the **GetFriendlyUrl** WSDL operation.
GetFriendlyUrlSoapOut	The response WSDL message for the **GetFriendlyUrl** WSDL operation.

3.1.4.15.1.1 GetFriendlyUrlSoapIn

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

The SOAP action value is:

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

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

3.1.4.15.1.2 GetFriendlyUrlSoapOut

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

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

3.1.4.15.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.

Element	Description
GetFriendlyUrl	The input data for the **GetFriendlyUrl** WSDL operation.
GetFriendlyUrlResponse	The result data for the **GetFriendlyUrl** WSDL operation.

3.1.4.15.2.1 GetFriendlyUrl

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

```
<xs:element name="GetFriendlyUrl" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="url" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

url: URL for obtaining the properties.

3.1.4.15.2.2 GetFriendlyUrlResponse

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

```
<xs:element name="GetFriendlyUrlResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetFriendlyUrlResult"
type="tns:FriendlyUrl"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetFriendlyUrlResult: Contains friendly URL properties specified in section 2.2.4.5

3.1.4.15.3 Complex Types

None.

3.1.4.15.4 Simple Types

None.

3.1.4.15.5 Attributes

None.

3.1.4.15.6 Groups

None.

3.1.4.15.7 Attribute Groups

None.

3.1.5 Timer Events

When the Wait timer expires, the protocol server MUST respond to the protocol client with a **WaitSoapOut** message, as specified in section 3.1.4.9.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples describe SOAP message pairings in hypothetical scenarios. As described in section 3.1, all operations are stateless and can be used independently.

4.1 Page and Page Layout Editing Suite

Client applications can leverage the functionality of publishing pages and page layouts to create new page layouts, disconnect publishing pages from page layouts to make customizations to a specific page, and reconnect pages to layouts to discard those customizations. Using the **CreatePageLayout**, **DisconnectPageLayout**, and **ReconnectPageLayout** operations in association with document editing functionality, the user can create a client-server publishing page editing suite.

Create a new page layout

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreatePageLayout xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <pageLayoutName>TestPageLayout</pageLayoutName>
      <associatedContentTypeId>0x010100D5C2F139516B419D801AC6C18942554D
</associatedContentTypeId>
      <title>Test Page Layout</title>
      <description>Layout used to demonstrate the functioning of the CreatePageLayout
operation</description>
      <siteUrl>http://www.contoso.com</siteUrl>
    </CreatePageLayout>
  </soap:Body>
</soap:Envelope>
```

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreatePageLayoutResponse       xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <CreatePageLayoutResult>http://www.contoso.com/PageLayouts/TestPageLayout.aspx
</CreatePageLayoutResult>
    </CreatePageLayoutResponse>
  </soap:Body>
</soap:Envelope>
```

Edit a publishing page and associate it with this page layout independent of the protocol. If customizations are desired, disconnect the publishing page from its page layout:

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DisconnectPageLayout       xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <pageUrl>http://www.contoso.com/Pages/TestPage.aspx</pageUrl>
    </DisconnectPageLayout>
  </soap:Body>
</soap:Envelope>
```

Response Body

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

Edit the publishing page independent of the protocol. If customizations are no longer desired, reconnect the publishing page with its page layout:

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ReconnectPageLayout       xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <pageUrl>http://www.contoso.com/Pages/TestPage.aspx</pageUrl>
    </ReconnectPageLayout>
  </soap:Body>
</soap:Envelope>
```

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ReconnectPageLayoutResponse      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <ReconnectPageLayoutResult>1,PageLayouts/TestPageLayout.aspx
</ReconnectPageLayoutResult>
    </ReconnectPageLayoutResponse>
  </soap:Body>
</soap:Envelope>
```

4.2 Translation Workflow

The processes of collecting and extracting content for translation and importing translated content can be automated by a remote protocol client that uses the **ExportObjects** and **ImportObjects** operations.

Extract translatable content

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ExportObjects xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <webUrl>http://office/ja-JP</webUrl>
    </ExportObjects>
  </soap:Body>
</soap:Envelope>
```

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ExportObjectsResponse       xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <ExportObjectsResult>ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuv
wxyz0123456789+/...</ExportObjectsResult>
    </ExportObjectsResponse>
  </soap:Body>
</soap:Envelope>
```

Translate content independent of to the protocol. Import translated content

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ImportObjects xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <siteUrl>http://office/ja-JP</siteUrl>
      <fileContent>ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123
456789+/...</fileContent>
    </ImportObjects>
  </soap:Body>
</soap:Envelope>
```

Response Body

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

Publish translated content for viewing independently of external to the protocol.

4.3 Publishing Dashboard

Protocol clients can retrieve scheduling, moderation, and versioning information about publishing objects from many different protocol servers to build current dashboard views.

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetObjectStatus xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <objectUrl>http://www.contoso.com/Pages/TestPage.aspx</objectUrl>
    </GetObjectStatus>
  </soap:Body>
</soap:Envelope>
```

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetObjectStatusResponse      xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <GetObjectStatusResult>
        <ObjectType>File</ObjectType>
        <Url>http://www.contoso.com/Pages/TestPage.aspx</Url>
        <PublishingUrl>http://office/en-us/TestPage.aspx</PublishingUrl>
        <LastMajorVersion>3.0</LastMajorVersion>
        <LastMajorModifiedTime>2008-02-01T19:54:04</LastMajorModifiedTime>
        <LastMinorVersion>0</LastMinorVersion>
        <LastMinorModifiedTime>0001-01-01T00:00:00</LastMinorModifiedTime>
        <ScheduledStartTime>1900-01-01T00:00:00Z</ScheduledStartTime>
        <ScheduledEndTime>2050-01-01T00:00:00Z</ScheduledEndTime>
        <ModerationStatus>Approved</ModerationStatus>
      </GetObjectStatusResult>
    </GetObjectStatusResponse>
  </soap:Body>
</soap:Envelope>
```

4.4 Script Sleep

The **Wait** operation is most commonly used by browser-based scripts associated with the synchronous XMLHTTPRequest (XHR) object to force the current script thread to stop execution for a specific amount of time.

Request Body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <Wait xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <millisecondsToWait>5000</millisecondsToWait >
    </Wait>
  </soap:Body>
</soap:Envelope>
```

Response Body

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

5 Security

5.1 Security Considerations for Implementers

Particular care is to be taken with authentication and authorization, or protocol clients using the **Wait** operation, because malicious client computers can use up protocol server resources by repeatedly running an operation with long wait parameters.

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

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
xmlns:s1="http://microsoft.com/wsdl/types/" 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:import namespace="http://microsoft.com/wsdl/types/" />
      <s:element name="GetObjectStatus">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="objectUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetObjectStatusResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetObjectStatusResult"
type="tns:PublishingObjectStatus" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PublishingObjectStatus">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="ObjectType"
type="tns:PublishingObjectType" />
          <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="PublishingUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="PathName" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LastMajorVersion" type="s:decimal" />
          <s:element minOccurs="1" maxOccurs="1" name="LastMajorModifiedTime"
type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="LastMinorVersion" type="s:decimal" />
          <s:element minOccurs="1" maxOccurs="1" name="LastMinorModifiedTime"
type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="ScheduledStartTime" type="s:dateTime"
/>
          <s:element minOccurs="1" maxOccurs="1" name="ScheduledEndTime" type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="ModerationStatus"
type="tns:SPModerationStatusType" />
        </s:sequence>
      </s:complexType>
      <s:simpleType name="PublishingObjectType">
        <s:restriction base="s:string">
          <s:enumeration value="ListItem" />
          <s:enumeration value="File" />
          <s:enumeration value="AccessDenied" />
          <s:enumeration value="FileNotFound" />
          <s:enumeration value="UnrecoverableFailure" />
          <s:enumeration value="Undefined" />
        </s:restriction>
      </s:simpleType>
      <s:simpleType name="SPModerationStatusType">
        <s:restriction base="s:string">
          <s:enumeration value="Approved" />
          <s:enumeration value="Denied" />
          <s:enumeration value="Pending" />
          <s:enumeration value="Draft" />
          <s:enumeration value="Scheduled" />
        </s:restriction>
      </s:simpleType>
      <s:element name="GetObjectStatusCollection">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="objectUrls" type="tns:ArrayOfString" 
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfString">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" 
type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetObjectStatusCollectionResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetObjectStatusCollectionResult" 
type="tns:ArrayOfPublishingObjectStatus" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfPublishingObjectStatus">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="PublishingObjectStatus" 
nillable="true" type="tns:PublishingObjectStatus" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetObjectStatusCollectionWithExclusions">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="objectUrls" type="tns:ArrayOfString" 
/>
            <s:element minOccurs="0" maxOccurs="1" name="thisPageUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetObjectStatusCollectionWithExclusionsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" 
name="GetObjectStatusCollectionWithExclusionsResult" type="tns:ArrayOfPublishingObjectStatus" 
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="Wait">
         <s:complexType>
          <s:sequence>
            <s:element name="millisecondsToWait" type="s:int"/>
          </s:sequence>
        </s:complexType>
      </s:element>
        <s:element name="WaitResponse">
         <s:complexType/>
      </s:element>
      <s:element name="ExportObjects">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="webUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ExportObjectsResponse">
```

```
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ExportObjectsResult"
type="s:base64Binary" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ImportObjects">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="siteUrl" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="fileContent" type="s:base64Binary"
/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ImportObjectsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="ImportObjectsResult"
type="s:boolean" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="CreatePageLayout">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="pageLayoutName" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="associatedContentTypeId"
type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="title" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="description" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="siteUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="CreatePageLayoutResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="CreatePageLayoutResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DisconnectPageLayout">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="pageUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DisconnectPageLayoutResponse">
        <s:complexType />
      </s:element>
      <s:element name="ReconnectPageLayout">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="pageUrl" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ReconnectPageLayoutResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="ReconnectPageLayoutResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetAllContentSlices">
        <s:complexType />
      </s:element>
      <s:element name="GetAllContentSlicesResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetAllContentSlicesResult" 
type="tns:ArrayOfContentSlice" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfContentSlice">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="ContentSlice" nillable="true" 
type="tns:ContentSlice" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ContentSlice">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Id" type="s1:guid" />
          <s:element minOccurs="0" maxOccurs="1" name="ChangeToken" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetFriendlyUrlBasedWebs">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="contentPartitionId" type="s1:guid" 
/>
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFriendlyUrlBasedWebsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetFriendlyUrlBasedWebsResult" 
type="tns:ArrayOfFriendlyUrlBasedWeb" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfFriendlyUrlBasedWeb">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="FriendlyUrlBasedWeb" 
nillable="true" type="tns:FriendlyUrlBasedWeb" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="FriendlyUrlBasedWeb">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SiteId" type="s1:guid" />
          <s:element minOccurs="1" maxOccurs="1" name="WebId" type="s1:guid" />
          <s:element minOccurs="0" maxOccurs="1" name="ServerRelativeWebUrl" type="s:string" 
/>
          <s:element minOccurs="1" maxOccurs="1" name="TypeOfChange" type="tns:ChangeType" />
        </s:sequence>
      </s:complexType>
      <s:simpleType name="ChangeType">
        <s:restriction base="s:string">
          <s:enumeration value="Update" />
          <s:enumeration value="Add" />
          <s:enumeration value="Delete" />
        </s:restriction>
      </s:simpleType>
      <s:element name="GetChangedFriendlyUrlBasedWebs">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="contentPartitionId" type="s1:guid" 
/>
            <s:element minOccurs="0" maxOccurs="1" name="changeToken" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetChangedFriendlyUrlBasedWebsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
name="GetChangedFriendlyUrlBasedWebsResult" type="tns:ArrayOfFriendlyUrlBasedWeb" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetAllFriendlyUrls">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="siteId" type="s1:guid" />
            <s:element minOccurs="1" maxOccurs="1" name="webId" type="s1:guid" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetAllFriendlyUrlsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetAllFriendlyUrlsResult"
type="tns:ArrayOfFriendlyUrl" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfFriendlyUrl">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="FriendlyUrl" nillable="true"
type="tns:FriendlyUrl" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="FriendlyUrl">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="ServerRelativeUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="HostName" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LastModified" type="s:dateTime" />
          <s:element minOccurs="1" maxOccurs="1" name="LastUpdated" type="s:dateTime" />
          <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyName" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyTitle" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyBrowserTitle"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyDescription"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyKeywords" type="s:string"
/>
          <s:element minOccurs="1" maxOccurs="1" name="SeoPropertyPriority" type="s:string"
/>
          <s:element minOccurs="0" maxOccurs="1" name="SeoPropertyChangeFrequency"
type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
name="SeoPropertyInternetSearchEngineInclusion" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Path" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="SecurityDescriptor"
type="s:base64Binary" />
          <s:element minOccurs="0" maxOccurs="1" name="CustomProperty" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="TypeOfChange" type="tns:ChangeType" />
          <s:element minOccurs="0" maxOccurs="1" name="UniqueId" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="GetChangedFriendlyUrls">
        <s:complexType>
```

```
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="siteId" type="s1:guid" />
            <s:element minOccurs="1" maxOccurs="1" name="webId" type="s1:guid" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetChangedFriendlyUrlsResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetChangedFriendlyUrlsResult" 
type="tns:ArrayOfFriendlyUrl" />
            <s:element minOccurs="0" maxOccurs="1" name="pagingToken" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFriendlyUrl">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="url" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFriendlyUrlResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetFriendlyUrlResult" 
type="tns:FriendlyUrl" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
    <s:schema elementFormDefault="qualified" 
targetNamespace="http://microsoft.com/wsdl/types/">
      <s:simpleType name="guid">
        <s:restriction base="s:string">
          <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-
9a-fA-F]{12}" />
        </s:restriction>
      </s:simpleType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="WaitSoapIn">
    <wsdl:part name="parameters" element="tns:Wait" />
  </wsdl:message>
  <wsdl:message name="WaitSoapOut">
    <wsdl:part name="parameters" element="tns:WaitResponse" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusSoapIn">
    <wsdl:part name="parameters" element="tns:GetObjectStatus" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusSoapOut">
    <wsdl:part name="parameters" element="tns:GetObjectStatusResponse" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusCollectionSoapIn">
    <wsdl:part name="parameters" element="tns:GetObjectStatusCollection" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusCollectionSoapOut">
    <wsdl:part name="parameters" element="tns:GetObjectStatusCollectionResponse" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusCollectionWithExclusionsSoapIn">
    <wsdl:part name="parameters" element="tns:GetObjectStatusCollectionWithExclusions" />
  </wsdl:message>
  <wsdl:message name="GetObjectStatusCollectionWithExclusionsSoapOut">
    <wsdl:part name="parameters" 
element="tns:GetObjectStatusCollectionWithExclusionsResponse" />
  </wsdl:message>
  <wsdl:message name="ExportObjectsSoapIn">
    <wsdl:part name="parameters" element="tns:ExportObjects" />
  </wsdl:message>
  <wsdl:message name="ExportObjectsSoapOut">
    <wsdl:part name="parameters" element="tns:ExportObjectsResponse" />
  </wsdl:message>
  <wsdl:message name="ImportObjectsSoapIn">
    <wsdl:part name="parameters" element="tns:ImportObjects" />
  </wsdl:message>
  <wsdl:message name="ImportObjectsSoapOut">
    <wsdl:part name="parameters" element="tns:ImportObjectsResponse" />
  </wsdl:message>
  <wsdl:message name="CreatePageLayoutSoapIn">
    <wsdl:part name="parameters" element="tns:CreatePageLayout" />
  </wsdl:message>
  <wsdl:message name="CreatePageLayoutSoapOut">
    <wsdl:part name="parameters" element="tns:CreatePageLayoutResponse" />
  </wsdl:message>
  <wsdl:message name="DisconnectPageLayoutSoapIn">
    <wsdl:part name="parameters" element="tns:DisconnectPageLayout" />
  </wsdl:message>
  <wsdl:message name="DisconnectPageLayoutSoapOut">
    <wsdl:part name="parameters" element="tns:DisconnectPageLayoutResponse" />
  </wsdl:message>
  <wsdl:message name="ReconnectPageLayoutSoapIn">
    <wsdl:part name="parameters" element="tns:ReconnectPageLayout" />
  </wsdl:message>
  <wsdl:message name="ReconnectPageLayoutSoapOut">
    <wsdl:part name="parameters" element="tns:ReconnectPageLayoutResponse" />
  </wsdl:message>
  <wsdl:message name="GetAllContentSlicesSoapIn">
    <wsdl:part name="parameters" element="tns:GetAllContentSlices" />
  </wsdl:message>
  <wsdl:message name="GetAllContentSlicesSoapOut">
    <wsdl:part name="parameters" element="tns:GetAllContentSlicesResponse" />
  </wsdl:message>
  <wsdl:message name="GetFriendlyUrlBasedWebsSoapIn">
    <wsdl:part name="parameters" element="tns:GetFriendlyUrlBasedWebs" />
  </wsdl:message>
  <wsdl:message name="GetFriendlyUrlBasedWebsSoapOut">
    <wsdl:part name="parameters" element="tns:GetFriendlyUrlBasedWebsResponse" />
  </wsdl:message>
  <wsdl:message name="GetChangedFriendlyUrlBasedWebsSoapIn">
    <wsdl:part name="parameters" element="tns:GetChangedFriendlyUrlBasedWebs" />
  </wsdl:message>
  <wsdl:message name="GetChangedFriendlyUrlBasedWebsSoapOut">
    <wsdl:part name="parameters" element="tns:GetChangedFriendlyUrlBasedWebsResponse" />
  </wsdl:message>
  <wsdl:message name="GetAllFriendlyUrlsSoapIn">
    <wsdl:part name="parameters" element="tns:GetAllFriendlyUrls" />
  </wsdl:message>
  <wsdl:message name="GetAllFriendlyUrlsSoapOut">
    <wsdl:part name="parameters" element="tns:GetAllFriendlyUrlsResponse" />
  </wsdl:message>
  <wsdl:message name="GetChangedFriendlyUrlsSoapIn">
    <wsdl:part name="parameters" element="tns:GetChangedFriendlyUrls" />
  </wsdl:message>
  <wsdl:message name="GetChangedFriendlyUrlsSoapOut">
    <wsdl:part name="parameters" element="tns:GetChangedFriendlyUrlsResponse" />
  </wsdl:message>
  <wsdl:message name="GetFriendlyUrlSoapIn">
    <wsdl:part name="parameters" element="tns:GetFriendlyUrl" />
  </wsdl:message>
  <wsdl:message name="GetFriendlyUrlSoapOut">
    <wsdl:part name="parameters" element="tns:GetFriendlyUrlResponse" />
  </wsdl:message>
  <wsdl:portType name="PublishingServiceSoap">
    <wsdl:operation name="Wait">
      <wsdl:input message="tns:WaitSoapIn" />
      <wsdl:output message="tns:WaitSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatus">
      <wsdl:input message="tns:GetObjectStatusSoapIn" />
      <wsdl:output message="tns:GetObjectStatusSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollection">
      <wsdl:input message="tns:GetObjectStatusCollectionSoapIn" />
      <wsdl:output message="tns:GetObjectStatusCollectionSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollectionWithExclusions">
      <wsdl:input message="tns:GetObjectStatusCollectionWithExclusionsSoapIn" />
      <wsdl:output message="tns:GetObjectStatusCollectionWithExclusionsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ExportObjects">
      <wsdl:input message="tns:ExportObjectsSoapIn" />
      <wsdl:output message="tns:ExportObjectsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ImportObjects">
      <wsdl:input message="tns:ImportObjectsSoapIn" />
      <wsdl:output message="tns:ImportObjectsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="CreatePageLayout">
      <wsdl:input message="tns:CreatePageLayoutSoapIn" />
      <wsdl:output message="tns:CreatePageLayoutSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="DisconnectPageLayout">
      <wsdl:input message="tns:DisconnectPageLayoutSoapIn" />
      <wsdl:output message="tns:DisconnectPageLayoutSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="ReconnectPageLayout">
      <wsdl:input message="tns:ReconnectPageLayoutSoapIn" />
      <wsdl:output message="tns:ReconnectPageLayoutSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetAllContentSlices">
      <wsdl:input message="tns:GetAllContentSlicesSoapIn" />
      <wsdl:output message="tns:GetAllContentSlicesSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrlBasedWebs">
      <wsdl:input message="tns:GetFriendlyUrlBasedWebsSoapIn" />
      <wsdl:output message="tns:GetFriendlyUrlBasedWebsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrlBasedWebs">
      <wsdl:input message="tns:GetChangedFriendlyUrlBasedWebsSoapIn" />
      <wsdl:output message="tns:GetChangedFriendlyUrlBasedWebsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetAllFriendlyUrls">
      <wsdl:input message="tns:GetAllFriendlyUrlsSoapIn" />
      <wsdl:output message="tns:GetAllFriendlyUrlsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrls">
      <wsdl:input message="tns:GetChangedFriendlyUrlsSoapIn" />
      <wsdl:output message="tns:GetChangedFriendlyUrlsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrl">
      <wsdl:input message="tns:GetFriendlyUrlSoapIn" />
      <wsdl:output message="tns:GetFriendlyUrlSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="PublishingServiceSoap" type="tns:PublishingServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetObjectStatus">
      <soap:operation 
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatus" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollection">
      <soap:operation 
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatusCollection" 
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollectionWithExclusions">
      <soap:operation 
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatusCollectionWithExclusi
ons" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Wait">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/Wait" 
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ExportObjects">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportObjects" 
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ImportObjects">
      <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/ImportObjects" 
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreatePageLayout">
      <soap:operation 
soapAction="http://schemas.microsoft.com/sharepoint/soap/CreatePageLayout" style="document" 
/>
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DisconnectPageLayout">
      <soap:operation 
soapAction="http://schemas.microsoft.com/sharepoint/soap/DisconnectPageLayout" 
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ReconnectPageLayout">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ReconnectPageLayout"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAllContentSlices">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAllContentSlices"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrlBasedWebs">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetFriendlyUrlBasedWebs"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrlBasedWebs">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangedFriendlyUrlBasedWebs"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAllFriendlyUrls">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAllFriendlyUrls" style="document"
/>
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrls">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangedFriendlyUrls"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrl">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetFriendlyUrl" 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="PublishingServiceSoap12" type="tns:PublishingServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="Wait">
      <soap12:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/Wait"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatus">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatus" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollection">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatusCollection"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetObjectStatusCollectionWithExclusions">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetObjectStatusCollectionWithExclusi
ons" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ExportObjects">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ExportObjects" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ImportObjects">
```

```
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ImportObjects" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreatePageLayout">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/CreatePageLayout" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DisconnectPageLayout">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/DisconnectPageLayout"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ReconnectPageLayout">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/ReconnectPageLayout"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAllContentSlices">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAllContentSlices"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrlBasedWebs">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetFriendlyUrlBasedWebs"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrlBasedWebs">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangedFriendlyUrlBasedWebs"
style="document" />
      <wsdl:input>
```

```
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAllFriendlyUrls">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetAllFriendlyUrls" style="document"
/>
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetChangedFriendlyUrls">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetChangedFriendlyUrls"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetFriendlyUrl">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/GetFriendlyUrl" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="PublishingService">
    <wsdl:port name="PublishingServiceSoap" binding="tns:PublishingServiceSoap">
      <soap:address location="http://localhost/_vti_bin/publishingservice.asmx" />
    </wsdl:port>
    <wsdl:port name="PublishingServiceSoap12" binding="tns:PublishingServiceSoap12">
      <soap12:address location="http://localhost/_vti_bin/publishingservice.asmx" />
    </wsdl:port>
  </wsdl:service>
</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 Designer 2007
- Microsoft SharePoint Designer 2010
- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Designer 2013
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016 Preview

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

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

<1> Section 2.2.4.4: This type is available only in SharePoint Designer 2013 and SharePoint Server 2013.

<2> Section 2.2.4.5: This type is available only in SharePoint Designer 2013 and SharePoint Server 2013.

<3> Section 2.2.4.6: This type is available only in SharePoint Designer 2013 and SharePoint Server 2013.

<4> Section 2.2.4.7: This type is available only in SharePoint Designer 2013 and SharePoint Server 2013.

<5> Section 3.1.1.3: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<6> Section 3.1.1.4: This concept is available only in SharePoint Server 2013.

<7> Section 3.1.1.5: This is available only in SharePoint Server 2013.

<8> Section 3.1.4: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<9> Section 3.1.4: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<10> Section 3.1.4: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<11> Section 3.1.4.1: Office SharePoint Server 2007 and SharePoint Server 2010 respect the specified file name if it ends with ".aspx". Otherwise, ".aspx" is appended to the specified file name.

<12> Section 3.1.4.4: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<13> Section 3.1.4.5: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<14> Section 3.1.4.9: This is available only in Office SharePoint Server 2007 and SharePoint Server 2010.

<15> Section 3.1.4.10: This operation is available only in SharePoint Server 2013.

<16> Section 3.1.4.11: This operation is available only in SharePoint Server 2013.

<17> Section 3.1.4.12: This operation is available only in SharePoint Server 2013.

<18> Section 3.1.4.13: This operation is available only in SharePoint Server 2013.

<19> Section 3.1.4.14: This operation is available only in SharePoint Server 2013.

<20> Section 3.1.4.15: This operation is available only in SharePoint Server 2013.

8 Change Tracking

This section identifies changes that were made to this document since the last release. 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.
- The 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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- Obsolete document removed.

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
7 Appendix B: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

9 Index