

[MS-PLSP]: Published Links 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.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Major	Updated and revised the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.5	Minor	Clarified the meaning of the technical content.
04/11/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	2.5	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	2.6	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
02/11/2013	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	2.7	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction	6
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	7
1.2.2 Informative References	7
1.3 Overview	8
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	9
1.7 Versioning and Capability Negotiation	9
1.8 Vendor-Extensible Fields	9
1.9 Standards Assignments	9
2 Messages	10
2.1 Transport	10
2.2 Common Message Syntax	10
2.2.1 Namespaces	10
2.2.2 Messages	11
2.2.3 Elements	11
2.2.4 Complex Types	11
2.2.5 Simple Types	11
2.2.6 Attributes	11
2.2.7 Groups	11
2.2.8 Attribute Groups	11
3 Protocol Details	12
3.1 Server Details	12
3.1.1 Abstract Data Model	12
3.1.2 Timers	12
3.1.3 Initialization	12
3.1.4 Message Processing Events and Sequencing Rules	12
3.1.4.1 GetLinks	12
3.1.4.1.1 Messages	13
3.1.4.1.1.1 GetLinksSoapIn	13
3.1.4.1.1.2 GetLinksSoapOut	13
3.1.4.1.2 Elements	13
3.1.4.1.2.1 GetLinks	14
3.1.4.1.2.2 GetLinksResponse	14
3.1.4.1.3 Complex Types	14
3.1.4.1.3.1 ArrayOfServerLink	14
3.1.4.1.3.2 ServerLink	14
3.1.4.1.4 Simple Types	16
3.1.4.1.5 Attributes	16
3.1.4.1.6 Groups	16
3.1.4.1.7 Attribute Groups	16
3.1.5 Timer Events	16
3.1.6 Other Local Events	16
4 Protocol Examples	17
5 Security	18

5.1 Security Considerations for Implementers.....	18
5.2 Index of Security Parameters	18
6 Appendix A: Full WSDL	19
7 Appendix B: Product Behavior	21
8 Change Tracking.....	22
9 Index	24

1 Introduction

The Published Links Web Service Protocol allows a protocol client to retrieve a list of potentially useful URLs and the descriptive details about each URL.

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

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
SOAP
SOAP action
SOAP body
SOAP fault
XML namespace

The following terms are defined in [\[MS-OFCGLOS\]](#):

data connection library
document library
list
personal site
portal site
profile site
site
Slide Library
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
XML namespace prefix
XML schema

The following terms are specific to this document:

Document Center: A document library template that is preconfigured to store a large quantity of documents.

Project Workspace site: A site that can be used to manage a project.

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

1.2 References

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

1.2.1 Normative References

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

[MS-TMPLDISC] Microsoft Corporation, "[Template Discovery Web Service Protocol](#)".

[RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., "Uniform Resource Locators (URL)", RFC 1738, December 1994, <http://www.ietf.org/rfc/rfc1738.txt>

[RFC1945] Berners-Lee, T., Fielding, R., and Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, <http://www.ietf.org/rfc/rfc1945.txt>

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, <http://ietf.org/rfc/rfc2045.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.ietf.org/rfc/rfc2616.txt>

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

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

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

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

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

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

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

1.2.2 Informative References

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

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

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

1.3 Overview

This protocol allows a protocol client to retrieve a list of **URLs** and descriptive details about each URL. A protocol client can choose to inspect the descriptive details to decide whether the URL is useful for its particular application. In all cases, "user" refers to the user account calling the service which has permissions to access the **site (2)**.

The types of URLs returned by default are:

- Sites (2) and **lists (1)** that an administrator has specified.
- Sites (2) where the user is a member.
- The user's **personal site**.
- The user's **profile site**.
- The user's **document libraries**.
- Document libraries from which document templates can be downloaded.

1.4 Relationship to Other Protocols

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

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

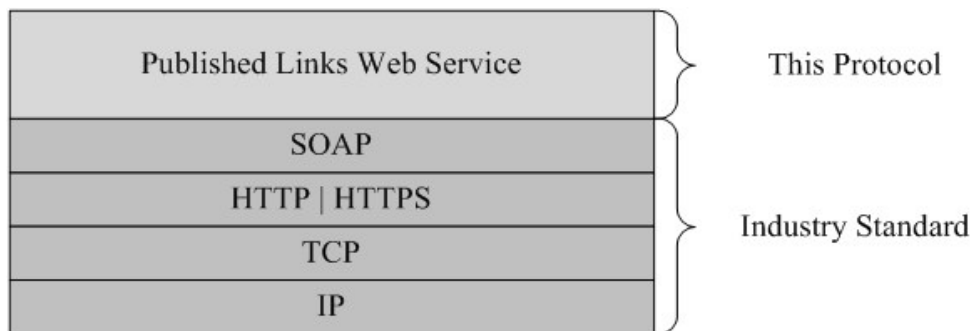


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

This protocol is applicable in scenarios where a protocol client can use a list of URLs. The protocol is intended for scenarios that include that of a protocol client displaying the list of URLs to the user calling the service. Thus, it would be an inappropriate use of this protocol for the protocol server to return more than 100 URLs to the client.

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 specified in section [2.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

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

2.2 Common Message Syntax

This section contains common definitions 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 protocol specifies and references **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this document associates an **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
http	http://schemas.xmlsoap.org/wsdl/http/	[RFC1945]
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
mime	http://schemas.xmlsoap.org/wsdl/mime/	[RFC2045]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
soapenc	http://schemas.xmlsoap.org/soap/encoding/	[SOAP1.2/1] [SOAP1.2/2]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] ↓ [XMLSCHEMA2] ↓
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
tm	http://microsoft.com/wsdl/mime/textMatching/	[WSDL]
tns	http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService	
(none)	http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService	

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

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

Except where specified, protocol clients SHOULD interpret the HTTP Status Codes that the protocol server returns as specified in [\[RFC2616\]](#), section 10.

This protocol allows protocol servers to use SOAP faults to notify protocol clients of application-level faults. Except where specified, these SOAP faults are not significant for interoperability; protocol clients can therefore interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults by using HTTP Status Codes or SOAP faults.

3.1 Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization 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.

The protocol server maintains a table of useful URLs and descriptive details for each URL.

3.1.2 Timers

None.

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
GetLinks	Retrieves a list of URLs and descriptive details about each URL.

3.1.4.1 GetLinks

This operation is used to retrieve a list of URLs relevant to the user calling the service and descriptive details about each URL. Relevant links are those that are either directly related to the user (like his personal site or personal document library) or that have been published by the site (2) administrator.

```
<wsdl:operation name="GetLinks">  
  <wsdl:input message="tns:GetLinksSoapIn" />  
</wsdl:operation>
```

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

The protocol client sends a **GetLinksSoapIn** request message and the protocol server MUST respond with a **GetLinksSoapOut** response message.

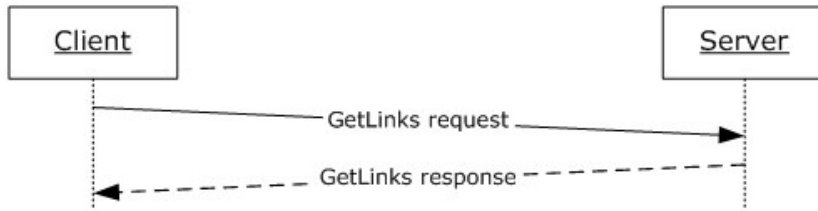


Figure 2: High-level sequence diagram for Published Links Web Service Protocol

3.1.4.1.1 Messages

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

Message	Description
GetLinksSoapIn	The request message for the GetLinks operation.
GetLinksSoapOut	The response message for the GetLinks operation.

3.1.4.1.1.1 GetLinksSoapIn

The request message for the **GetLinks** operation. The **SOAP action** value of the message is defined as:

```
http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService/GetLinks
```

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

3.1.4.1.1.2 GetLinksSoapOut

The response message for the **GetLinks** operation.

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

3.1.4.1.2 Elements

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

Element	Description
GetLinks	Used to issue the GetLinks request.
GetLinksResponse	The response to the GetLinks request.

3.1.4.1.2.1 GetLinks

The **GetLinks** element is used to issue the **GetLinks** request.

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

This element MUST NOT contain child elements.

3.1.4.1.2.2 GetLinksResponse

The **GetLinksResponse** element is returned as the response to the **GetLinks** request.

```
<s:element name="GetLinksResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="GetLinksResult" type="tns:ArrayOfServerLink" minOccurs="0"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetLinksResult: List of URLs and descriptive details for each URL.

3.1.4.1.3 Complex Types

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

Complex type	Description
ArrayOfServerLink	An array of ServerLink elements.
ServerLink	A URL and descriptive details about the URL.

3.1.4.1.3.1 ArrayOfServerLink

The **ArrayOfServerLink** complex type represents an array of **ServerLink** elements.

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

ServerLink: See section [3.1.4.1.3.2](#).

3.1.4.1.3.2 ServerLink

The **ServerLink** complex type represents a URL and descriptive details about the URL.

```

<s:complexType name="ServerLink">
  <s:sequence>
    <s:element name="Title" type="s:string" minOccurs="1" maxOccurs="1"/>
    <s:element name="Url" type="s:string" minOccurs="1" maxOccurs="1"/>
    <s:element name="LinkType" type="s:long" minOccurs="1" maxOccurs="1"/>
    <s:element name="IsMember" type="s:boolean" minOccurs="1" maxOccurs="1"/>
    <s:element name="IsPublished" type="s:boolean" minOccurs="1" maxOccurs="1"/>
  </s:sequence>
</s:complexType>

```

Title: Title of the location corresponding to the URL. The value MUST NOT be null or empty. The string MUST be less than 255 characters in length and MUST NOT contain any of the following characters: \ / : * ? " < > |.

Url: The URL of the location. The value MUST NOT be null or empty and MUST conform to the URL format, as specified in [\[RFC1738\]](#).

LinkType: The type of the URL. The value MUST be set to one of the following values:

- 0x0000000000000001: site (2)
- 0x0000000000000002: personal site
- 0x0000000000000004: **portal site**
- 0x0000000000000008: **Project Workspace site**
- 0x0000000001000000: **Document Center**
- 0x0000000002000000: document library
- 0x0000000004000000: **data connection library**
- 0x0000000008000000: **Slide Library**
- 0x0100000000000000: profile site
- 0x0200000000000000: Personal Documents library
- 0x1000000002000000: document library template source
- 0x1000000010000000: asset library template source
- 0x1000000020000000: process repository template source

IsMember: Indicates whether the user credentials being used to call the service are listed in the Members group of the site (2) on which the service is being called.

IsPublished: This element has no meaning and MUST be ignored by the protocol client.

Additional Requirements:

- Three "template source" values are shown in the preceding list: **document library template source**, **asset library template source**, and **process repository template source**. If the protocol server sets **LinkType** to one of these values, the server MUST support the Template Discovery Web Service Protocol [\[MS-TMPLDISC\]](#).

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.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

A protocol client might construct the following WSDL message to retrieve a list of URLs.

```
<?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>
    <GetLinks
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService" />
  </soap:Body>
</soap:Envelope>
```

The protocol server returns the list of URLs in the form of the following WSDL message:

```
<?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>
    <GetLinksResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService">
      <GetLinksResult>
        <ServerLink>
          <Title>My Site</Title>
          <Url>http://contoso:80/personal/bob/</Url>
          <LinkType>2</LinkType>
          <IsMember>true</IsMember>
          <IsPublished>true</IsPublished>
        </ServerLink>
        <ServerLink>
          <Title>Profile Site</Title>
          <Url>http://contoso:80/mysite/Person.aspx?user=</Url>
          <LinkType>72057594037927936</LinkType>
          <IsMember>>false</IsMember>
          <IsPublished>true</IsPublished>
        </ServerLink>
        <ServerLink>
          <Title>Shared Documents</Title>
          <Url>http://contoso:80/personal/bob/Shared%20Documents</Url>
          <LinkType>144115188075855872</LinkType>
          <IsMember>>false</IsMember>
          <IsPublished>true</IsPublished>
        </ServerLink>
        <ServerLink>
          <Title>Contoso Document Library</Title>
          <Url>http://contoso:80/Documents</Url>
          <LinkType>1152921504640401408</LinkType>
          <IsMember>>false</IsMember>
          <IsPublished>>false</IsPublished>
        </ServerLink>
      </GetLinksResult>
    </GetLinksResponse>
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

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://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService"
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://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService">
      <s:element name="GetLinks">
        <s:complexType />
      </s:element>
      <s:element name="GetLinksResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetLinksResult"
type="tns:ArrayOfServerLink" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfServerLink">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="ServerLink"
nillable="true" type="tns:ServerLink" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ServerLink">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Title" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="Url" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="LinkType" type="s:long" />
          <s:element minOccurs="1" maxOccurs="1" name="IsMember" type="s:boolean" />
          <s:element minOccurs="1" maxOccurs="1" name="IsPublished"
type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetLinksSoapIn">
    <wsdl:part name="parameters" element="tns:GetLinks" />
  </wsdl:message>
  <wsdl:message name="GetLinksSoapOut">
    <wsdl:part name="parameters" element="tns:GetLinksResponse" />
  </wsdl:message>
  <wsdl:portType name="PublishedLinksServiceSoap">
    <wsdl:operation name="GetLinks">
      <wsdl:input message="tns:GetLinksSoapIn" />
      <wsdl:output message="tns:GetLinksSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

```

</wsdl:portType>
<wsdl:binding name="PublishedLinksServiceSoap"
  type="tns:PublishedLinksServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <wsdl:operation name="GetLinks">
    <soap:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService/Get
      Links" 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="PublishedLinksServiceSoap12"
  type="tns:PublishedLinksServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <wsdl:operation name="GetLinks">
    <soap12:operation
      soapAction="http://microsoft.com/webservices/SharePointPortalServer/PublishedLinksService/Get
      Links" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>

```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- The 2007 Microsoft Office system
- Microsoft Office 2010 suites
- Microsoft Office 2013
- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Workspace 2010
- Microsoft Visio 2010
- Microsoft Visual Studio 2010

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.

8 Change Tracking

This section identifies changes that were made to the [MS-PLSP] protocol document between the February 2013 and July 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.3 Overview	Appended "(2)" to glossary "site".	N	Content updated.
1.5 Prerequisites/Preconditions	Appended "(2)" to glossary "site".	N	Content updated.
3.1.4.1 GetLinks	Appended "(2)" to glossary "site".	N	Content updated.
3.1.4.1.1.2 GetLinksSoapOut	Removed the description for soap action value from section 3.1.4.1.1.2 GetLinksSoapOut.	N	Content removed.
3.1.4.1.3.2 ServerLink	Appended "(2)" to glossary "site".	N	Content updated.

9 Index

A

Abstract data model
 [server](#) 12
[Applicability](#) 9
[Attribute groups](#) 11
[Attributes](#) 11

C

[Capability negotiation](#) 9
[Change tracking](#) 22
Client
 [overview](#) 12
[Complex types](#) 11
 server
 [ArrayOfServerLink](#) 14
 [ServerLink](#) 14

D

Data model - abstract
 [server](#) 12

E

Elements
 server
 [GetLinks](#) 14
 [GetLinksResponse](#) 14
Events
 [local - server](#) 16
 [timer - server](#) 16
[Example](#) 17

F

[Fields - vendor-extensible](#) 9
[Full WSDL](#) 19

G

[GetLinks operation](#) 12
 [attribute groups](#) 16
 [attributes](#) 16
 [complex types](#) 14
 [ArrayOfServerLink](#) 14
 [ServerLink](#) 14
 [elements](#) 13
 [GetLinks](#) 14
 [GetLinksResponse](#) 14
 [groups](#) 16
 [messages](#) 13
 [GetLinksSoapIn](#) 13
 [GetLinksSoapOut](#) 13
 [simple types](#) 16
[Glossary](#) 6
[Groups](#) 11

I

[Implementer - security considerations](#) 18
[Index of security parameters](#) 18
[Informative references](#) 7
Initialization
 [server](#) 12
[Introduction](#) 6

L

Local events
 [server](#) 16

M

Message processing
 [server](#) 12
Messages
 [attribute groups](#) 11
 [attributes](#) 11
 [complex types](#) 11
 [elements](#) 11
 [enumerated](#) 11
 [groups](#) 11
 [namespaces](#) 10
 server
 [GetLinksSoapIn](#) 13
 [GetLinksSoapOut](#) 13
 [simple types](#) 11
 [syntax](#) 10
 [transport](#) 10

N

[Namespaces](#) 10
[Normative references](#) 7

O

Operations
 [GetLinks](#) 12
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 18
[Preconditions](#) 8
[Prerequisites](#) 8
[Product behavior](#) 21

R

[References](#) 6
 [informative](#) 7
 [normative](#) 7
[Relationship to other protocols](#) 8

S

Security

[implementer considerations](#) 18
[parameter index](#) 18

Sequencing rules

[server](#) 12

Server

[abstract data model](#) 12
[GetLinks operation](#) 12
[initialization](#) 12
[local events](#) 16
[message processing](#) 12
[overview](#) 12
[sequencing rules](#) 12
[timer events](#) 16
[timers](#) 12

[Simple types](#) 11

[Standards assignments](#) 9

Syntax

[messages - overview](#) 10

T

Timer events

[server](#) 16

Timers

[server](#) 12

[Tracking changes](#) 22

[Transport](#) 10

Types

[complex](#) 11

[simple](#) 11

V

[Vendor-extensible fields](#) 9

[Versioning](#) 9

W

[WSDL](#) 19