

[MS-BCSDWPS]:

Business Connectivity Services Deployment Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Major	Updated and revised the technical content
6/7/2010	1.03	Editorial	Revised and edited the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	1.5	Minor	Clarified the meaning of the technical content.
4/11/2012	1.5	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	1.5	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	1.5	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.6	Minor	Clarified the meaning of the technical content.
2/11/2013	1.6	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.7	Minor	Clarified the meaning of the technical content.
11/18/2013	1.7	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.7	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.8	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
7/31/2014	1.8	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.8	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	2.0	Major	Significantly changed the technical content.
2/26/2016	3.0	Major	Significantly changed the technical content.
7/15/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/20/2017	3.1	Minor	Clarified the meaning of the technical content.
7/24/2018	4.0	Major	Significantly changed the technical content.
10/1/2018	5.0	Major	Significantly changed the technical content.
4/22/2021	6.0	Major	Significantly changed the technical content.
7/20/2021	7.0	Major	Significantly changed the technical content.
8/17/2021	8.0	Major	Significantly changed the technical content.
10/5/2021	8.0	None	No changes to the meaning, language, or formatting of the technical content.
2/15/2022	8.0	None	No changes to the meaning, language, or formatting of the technical content.
4/16/2024	9.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	9
1.2.1	Normative References	9
1.2.2	Informative References	9
1.3	Overview	10
1.4	Relationship to Other Protocols	10
1.5	Prerequisites/Preconditions	10
1.6	Applicability Statement	11
1.7	Versioning and Capability Negotiation	11
1.8	Vendor-Extensible Fields	11
1.9	Standards Assignments.....	11
2	Messages.....	12
2.1	Transport	12
2.2	Common Message Syntax	12
2.2.1	Namespaces	12
2.2.2	Messages.....	12
2.2.3	Elements	12
2.2.4	Complex Types.....	13
2.2.5	Simple Types	13
2.2.6	Attributes	13
2.2.7	Groups	13
2.2.8	Attribute Groups.....	13
2.2.9	Common Data Structures	13
3	Protocol Details	14
3.1	Server Details.....	14
3.1.1	Abstract Data Model.....	14
3.1.2	Timers	15
3.1.3	Initialization.....	15
3.1.4	Message Processing Events and Sequencing Rules	15
3.1.4.1	CreateProfilePage.....	15
3.1.4.1.1	Messages	16
3.1.4.1.1.1	CreateProfilePageSoapIn.....	16
3.1.4.1.1.2	CreateProfilePageSoapOut.....	17
3.1.4.1.2	Elements.....	17
3.1.4.1.2.1	CreateProfilePage.....	17
3.1.4.1.2.2	CreateProfilePageResponse	17
3.1.4.1.3	Complex Types	17
3.1.4.1.4	Simple Types	18
3.1.4.1.5	Attributes	18
3.1.4.1.6	Groups.....	18
3.1.4.1.7	Attribute Groups.....	18
3.1.4.2	GetPackages	18
3.1.4.2.1	Messages	19
3.1.4.2.1.1	GetPackagesSoapIn	19
3.1.4.2.1.2	GetPackagesSoapOut	19
3.1.4.2.2	Elements.....	20
3.1.4.2.2.1	GetPackages	20
3.1.4.2.2.2	GetPackagesResponse	20
3.1.4.2.3	Complex Types	20
3.1.4.2.3.1	ArrayOfString	21
3.1.4.2.3.2	ArrayOfPackageInfo	21
3.1.4.2.3.3	PackageInfo	21

3.1.4.2.4	Simple Types	22
3.1.4.2.5	Attributes	22
3.1.4.2.6	Groups	22
3.1.4.2.7	Attribute Groups.....	22
3.1.4.3	UpsizeVirtualList	22
3.1.4.3.1	Messages	23
3.1.4.3.1.1	UpsizeVirtualListSoapIn	23
3.1.4.3.1.2	UpsizeVirtualListSoapOut	24
3.1.4.3.2	Elements.....	24
3.1.4.3.2.1	UpsizeVirtualList	24
3.1.4.3.2.2	UpsizeVirtualListResponse.....	24
3.1.4.3.3	Complex Types	24
3.1.4.3.4	Simple Types	24
3.1.4.3.5	Attributes	25
3.1.4.3.6	Groups.....	25
3.1.4.3.7	Attribute Groups.....	25
3.1.5	Timer Events.....	25
3.1.6	Other Local Events.....	25
4	Protocol Examples	26
4.1	Retrieve a List of Solution Package Locations Associated with a Set of External Lists	26
4.2	Create a Profile Page for an Existing External Content Type	26
5	Security	28
5.1	Security Considerations for Implementers	28
5.2	Index of Security Parameters	28
6	Appendix A: Full WSDL	29
7	Appendix B: Product Behavior	32
8	Change Tracking.....	34
9	Index.....	35

1 Introduction

The Business Connectivity Services Deployment Web Service Protocol provides utility functions that can be used in the creation of development or administration tools and applications that interact with Business Connectivity Services solutions.

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

1.1 Glossary

This document uses the following terms:

action: The smallest unit of work in a workflow system. An action can contain one or more tasks that define work that actors need to do. Actions are deployed and registered in the workflow system to be activated by protocol client users.

ActionParameter: A type of **MetadataObject** that defines how to parameterize the URL of an **Action** with specific data about an **EntityInstance**. ActionParameters are contained by Actions.

certificate: A certificate is a collection of attributes and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [\[RFC3280\]](#) and [\[X509\]](#) sections 7 and 8.

creator: The user who creates protected content.

curly braced GUID string: The string representation of a 128-bit globally unique identifier (**GUID**) using the form {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}, where X denotes a hexadecimal digit. The string representation between the enclosing braces is the standard representation of a GUID as described in [\[RFC4122\]](#) section 3. Unlike a GUIDString, a curly braced GUID string includes enclosing braces.

deployment package: A collection of files that can be used to deploy and manage customizations, such as add-ins, to a computer. It consists of an application manifest, a deployment manifest, and related package files.

external content type: A type of DataClass object that is stored in a line-of-business (LOB) system and whose instances have a persistent EntityInstanceId. Also referred to as Entity.

external list: A container that is within a SharePoint site and that references a set of EntityInstances that are sourced from a line-of-business (LOB) system. It has a customizable schema that is composed of one or more Fields.

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

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.

form: A document with a set of controls into which users can enter information. Controls on a form can be bound to elements in the data source of the form, such as fields and groups. See also [bind](#).

form template: A file or set of files that defines the data structure, appearance, and behavior of a [form](#).

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).

GUIDString: A [GUID](#) in the form of an ASCII or Unicode string, consisting of one group of 8 hexadecimal digits, followed by three groups of 4 hexadecimal digits each, followed by one group of 12 hexadecimal digits. It is the standard representation of a GUID, as described in [\[RFC4122\]](#) section 3. For example, "6B29FC40-CA47-1067-B31D-00DD010662DA". Unlike a curly braced GUID string, a GUIDString is not enclosed in braces.

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 [\[RFC5246\]](#).

identifier: A string value that is used to uniquely identify a component of the CSDL and that is of type SimpleIdentifier.

LobSystem: A type of [MetadataObject](#) that represents a specific version of a line-of-business (LOB) system. An LOB system can be a database or a web service.

MetadataObject: An abstract data structure that consists of a set of attributes that represent a LobSystem, LobSystemInstance, DataClass, Entity, Method, MethodInstance, Parameter, TypeDescriptor, Identifier, FilterDescriptor, [Action](#), [ActionParameter](#), or Association.

profile page: An ASP.NET webpage that includes Web Part controls that display data from a line-of-business (LOB) system. By default, it contains a single Web Part that displays the details of a single, primary EntityInstance. If the primary Entity has associated Entities and is the source of an Association, the profile page also contains a related Web Part for each associated Entity, and each related Web Part displays EntityInstances that are associated with the primary EntityInstance.

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.

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

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

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

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

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

Updater: A type of MethodInstance that can be called to update an EntityInstance identified by a specified EntityInstanceId. The set of Fields that is required to update the EntityInstance is referred to as the Updater View.

view: See form view (Microsoft InfoPath), list view (SharePoint Products and Technologies), or **View** (Microsoft Business Connectivity Services).

Web Part Page: An ASP.NET webpage that includes Web Part controls that enable users to customize the page, such as specifying which information to display. Referred to as Web Parts Page in Microsoft SharePoint Foundation 2010.

web service: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as **HTTP**, Simple Mail Transfer Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

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

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

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

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

1.2 References

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

1.2.1 Normative References

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

[MS-BCSDPFFS] Microsoft Corporation, "[Business Connectivity Services Deployment Package File Format](#)".

[MS-IPFF2] Microsoft Corporation, "[InfoPath Form Template Format Version 2](#)".

[MS-OSCO] Microsoft Corporation, "[Office Server ClickOnce Manifest Structure](#)".

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

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

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

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

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

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <https://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, <https://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

1.2.2 Informative References

[IANAPORT] IANA, "Service Name and Transport Protocol Port Number Registry", <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>

1.3 Overview

This protocol defines three operations that allow a protocol client to perform the following tasks:

- Create a **profile page** for an existing **external content type**
- Retrieve a list of **deployment package** locations that are associated with a set of server-side objects
- Associate a **form** with an **external list**

Profile pages represent an instance of an external content type. For example, a customer record in a business application or in search results. The protocol client can use the profile page creation operation to initiate the generation of a profile page for a specific business data object. Typical clients for this protocol are administrative tools for the protocol server or design tools for data modeling.

In some scenarios, it is desirable to associate with a server object deployment packages that represent additional capabilities to be used on a client. The process that is used to retrieve a list of deployment packages is designed for this scenario. This process allows the protocol client to send a collection of identifiers for external lists to the protocol server and to receive from the protocol server a list of deployment package locations. A typical scenario for using this protocol is a Web application that displays business data that is also useful on clients. One example of such a client is a contact management application that provides data caching to support use cases in which a network connection is only occasional.

To support the advanced data validation or UI customization of list data, it is often desirable to replace standard forms with custom forms. This protocol enables a protocol client to associate a custom form at a known location with a list on the protocol server. This protocol also enables a protocol client to map the create, view, and update actions of the list to the actions that the custom form supplies.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [\[SOAP1.1\]](#), [\[SOAP1.2-1/2007\]](#) and [\[SOAP1.2-2/2007\]](#). It transmits those messages by using **HTTP**, as described in [\[RFC2616\]](#), or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [\[RFC2818\]](#).

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

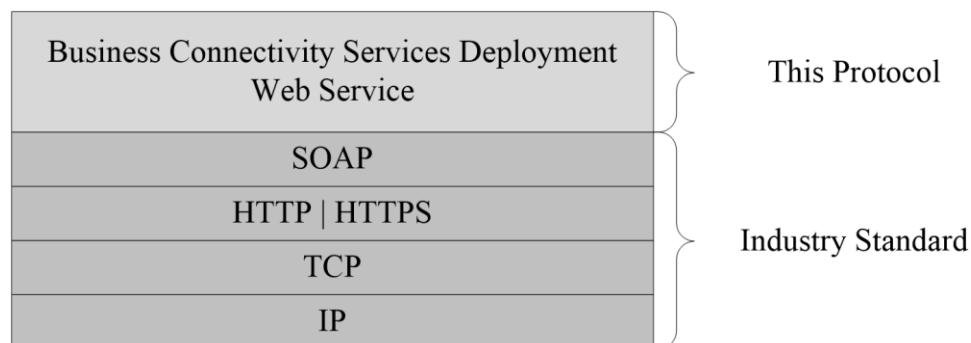


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

This protocol assumes that the underlying protocols perform authentication.

The **profile page** creation and the **form** association processes include operations to create and modify server objects. This protocol requires the protocol client to have the permissions that enable the creation or modification of server objects.

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

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

This protocol uses standard Internet Assigned Numbers Authority (IANA) port assignments for **HTTP**, HTTP Alternate, SOCKS, Secure Sockets Layer (SSL), and Secure Socket Tunneling Protocol (SSTP). These standard port assignments use IANA-assigned ports, as shown in the following table.

Parameter	Value	Reference
IANA-assigned port for HTTP	80	[IANAPORT]
IANA-assigned port for HTTP Alternate	8080	[IANAPORT]
IANA-assigned port for SOCKS	1080	[IANAPORT]
IANA-assigned port for SSL	443	[IANAPORT]
IANA-assigned port for SSTP	2492	[IANAPORT]

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

2.1 Transport

The protocol **MUST** support **SOAP** over **HTTP**. To help secure communication with protocol clients, the protocol **SHOULD** additionally support SOAP over **HTTPS**.

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

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 [\[XMLSCHEMA1/2\]](#) and [\[XMLSCHEMA2/2\]](#), and **WSDL**, as specified in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
http	http://schemas.xmlsoap.org/wsdl/http/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2-1/2007] [SOAP1.2-2/2007]
tns	http://microsoft.com/webservices/SharePointPortalServer/ObaWS/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2] [XMLSCHEMA2/2]

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.

2.2.9 Common Data Structures

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

3 Protocol Details

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL might specify differences that reflect actual Microsoft product behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, and **present**.

This protocol operates between a protocol client and a protocol server. The protocol server does not retain any state.

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 are returned by the protocol server as specified in [\[RFC2616\]](#) section 10 Status Code Definitions.

This protocol allows protocol servers to notify protocol clients of application-level faults by 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 to notify protocol clients of authorization faults, either by using HTTP status codes or by using SOAP faults as specified earlier in this section.

3.1 Server Details

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.

The protocol server maintains the following data:

- A **site** with which the **Web service** is associated
- A collection of **folders**
- A collection of **form templates** that conform to the form template format specified in [\[MS-IPFF2\]](#)
- A collection of **Web Part Pages**
- A collection of **external lists**
- A collection of actions for each external list
- A collection of **views** for each external list
- A URL that specifies the site that hosts **profile pages**. All profile pages will be generated on this site.

The protocol server maintains a collection of the following **MetadataObject** artifacts:

- A collection of **LobSystem**
- A collection of **LobSystem** properties associated with each **LobSystem**
- A collection of **external content types**, each of which is contained in a **LobSystem**

The protocol server has to follow the following rules:

- Each external content type is identified by the combination of namespace and name.
- Each external content type has a version number.
- An external content type can have another external content type associated with it.
- Each external list has a **GUID**.
- Each external list is associated with a folder, and each folder is not associated with more than one external list.
- Each folder associated with an external list has a folder named Item, and each folder named Item can have a folder named Form.
- Each form template is contained in an Item folder.
- Each external list is based on an external content type.
- Each external list has three special actions for creating, viewing, and updating the items of the list, and each of the actions has a URL that points to one of the Web Part Pages.

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 operations as defined by this specification.

Operation	Description
CreateProfilePage	The CreateProfilePage operation generates a profile page for a particular external content type .
GetPackages	The GetPackages operation returns the information about the deployment packages that are associated with a list of external lists .
UpsizeVirtualList	The UpsizeVirtualList operation generates a Web Part Page for viewing an item of an external list based on the specified form template and updates the viewing action of the external list to point to the Web Part Page.

3.1.4.1 CreateProfilePage

The **CreateProfilePage** operation generates a **profile page** for a particular **external content type**.

The following is the **WSDL** port type specification of the **CreateProfilePage WSDL operation**.

```
<wsdl:operation name="CreateProfilePage" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CreateProfilePageSoapIn"/>
  <wsdl:output message="tns:CreateProfilePageSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **CreateProfilePageSoapIn** request **WSDL message**. The protocol server MUST respond with a **CreateProfilePageSoapOut** response message, as follows:

- The protocol client MUST specify the namespace and the name of the external content type for which the profile page will be created.
- The protocol server SHOULD [<1>](#) generate a profile page for the particular external content type with an implementation specific URL [<2>](#).

If the protocol server returns a **SOAP fault** as specified in [\[SOAP1.1\]](#) section 4.4, the SOAP fault MUST include a fault code value of Server as specified in [\[SOAP1.1\]](#) section 4.4.1. The protocol server MUST return this SOAP fault when any of the following occur:

- The namespace or the name of the external content type is not specified.
- The namespace or the name of the external content type is an empty string.
- The namespace or the name of the external content type has more than 255 Unicode characters.
- The external content type of the particular namespace and name cannot be found.
- The **site** that hosts profile pages is not specified.
- The site that hosts profile pages does not exist. [<3>](#)
- The site that hosts profile pages does not have access to the external content type specified by the namespace and name.
- The user performing the client protocol does not have permission to create, edit, and remove a page on the site that hosts profile pages.

3.1.4.1.1 Messages

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

Message	Description
CreateProfilePageSoapIn	The request WSDL message for the CreateProfilePage WSDL operation .
CreateProfilePageSoapOut	The response WSDL message for the CreateProfilePage WSDL operation .

3.1.4.1.1.1 CreateProfilePageSoapIn

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

The **SOAP action** value is:

```
http://microsoft.com/webservices/SharePointPortalServer/ObaWS/CreateProfilePage
```


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

3.1.4.1.1.2 CreateProfilePageSoapOut

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

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

3.1.4.1.2 Elements

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

Element	Description
CreateProfilePage	The input data for the CreateProfilePage WSDL operation .
CreateProfilePageResponse	The result data for the CreateProfilePage WSDL operation .

3.1.4.1.2.1 CreateProfilePage

The **CreateProfilePage** element specifies the input data for the **CreateProfilePage WSDL operation**.

```
<xs:element name="CreateProfilePage" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="entityNamespace" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="entityName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="allowOverwrite" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

entityNamespace: The namespace of the **external content type** of which the **profile page** will be created. The value of this element **MUST** have at least 1 and no more than 255 Unicode characters. This element **MUST** be presented.

entityName: The name of the external content type of which the profile page will be created. The value of this element **MUST** have at least 1 and no more than 255 Unicode characters. This element **MUST** be presented.

allowOverwrite: The value of this element **MUST** be **true** or **false**, specifying whether the profile page to be created overwrites any previously created profile page. The value of this element **MUST** be ignored by the protocol server.

3.1.4.1.2.2 CreateProfilePageResponse

The **CreateProfilePageResponse** element specifies the result data for the **CreateProfilePage WSDL operation**.

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

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 GetPackages

The **GetPackages** operation returns the information about the **deployment packages** that are associated with a list of **external lists**.

The following is the **WSDL** port type specification of the **GetPackages WSDL operation**.

```
<wsdl:operation name="GetPackages" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetPackagesSoapIn"/>
  <wsdl:output message="tns:GetPackagesSoapOut"/>
</wsdl:operation>
```

If the deployment package of an external list does not exist or the external list has been modified since the deployment package was generated, this operation **MUST** also generate a new deployment package for the external list. The deployment packages **MUST** conform to the format specified in both [\[MS-BCSDPFFS\]](#) and [\[MS-OSCO\]](#).

The protocol client sends a **GetPackagesSoapIn** request **WSDL message**. The protocol server **SHOULD** [<4>](#) respond with a **GetPackagesSoapOut** response WSDL message. The sequence is as follows:

- The protocol client **MUST** specify the identifiers of a collection of external lists for which the deployment package information will be returned.
- If the protocol server determines that the deployment package of an external list does not exist, the protocol server **MUST** generate a deployment package for the external list.
- If there is a deployment package for an external list, the protocol server **MUST** check whether changes were made to the external list after the deployment package was generated. When any of the following changes occur, the protocol server **MUST** generate a new deployment package:
 - A **file** is added to the Item or Form **folder** that is associated with the external list.
 - A file contained in the Item or Form folder that is associated with the external list is modified.
 - A **view** of the external list is changed.
 - A view is created for the external list.
 - A view of the external list is removed.

- The version of the **external content type** of the external list is changed.
- The version of an external content type that is associated with the external content type of the list is changed.
- A new external content type is associated with the external content type of the list.
- The value of the "LastModifiedTimestamp" property of the **LobSystem** that contains either the external content type of the list or its associated external content types is changed.
- The protocol server MUST return the information about the deployment packages that correspond to the external lists.

If the protocol server returns a **SOAP fault** as specified in [SOAP1.1] section 4.4, the SOAP fault MUST include a fault code value of Server as specified in [SOAP1.1] section 4.4.1. The protocol server MUST return this SOAP fault when any of the following occur:

- The protocol server does not support this operation.
- The list of identifiers is not specified.
- The list of identifiers is empty.
- The identifiers are not a **GUIDString** or a **curly braced GUID string**.
- The external list that is specified by any identifier does not exist on the **site** that is associated with the **Web service**.
- The site that is associated with this Web service allows anonymous access.

If the user performing the client protocol does not have read permission on the site or the external lists, the server MUST return **HTTP** status code 401 as specified in [RFC2616] section 10.

3.1.4.2.1 Messages

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

Message	Description
GetPackagesSoapIn	The request WSDL message for the GetPackages WSDL operation .
GetPackagesSoapOut	The response WSDL message for the GetPackages WSDL operation .

3.1.4.2.1.1 GetPackagesSoapIn

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

The **SOAP action** value is:

```
http://microsoft.com/webservices/SharePointPortalServer/ObaWS/GetPackages
```

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

3.1.4.2.1.2 GetPackagesSoapOut

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

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

3.1.4.2.2 Elements

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

Element	Description
GetPackages	The input data for the GetPackages WSDL operation .
GetPackagesResponse	The result data for the GetPackages WSDL operation .

3.1.4.2.2.1 GetPackages

The **GetPackages** element specifies the input data for the **GetPackages WSDL operation**.

```
<xs:element name="GetPackages" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="listIds" type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="clientVersion" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

listIds: The list of **external list** identifiers of which the **deployment package** information will be returned. Each identifier is a **GUIDString** or **curly braced GUID string**. This element **MUST** be presented.

clientVersion: Specifies the version of client where the package will be installed. Supported version numbers are 14.0.0.0 and 15.0.0.0.

3.1.4.2.2.2 GetPackagesResponse

The **GetPackagesResponse** element specifies the result data for the **GetPackages WSDL operation**.

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

GetPackagesResult: An array of complex type **PackageInfo** that represents **deployment packages**.

3.1.4.2.3 Complex Types

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

Complex type	Description
ArrayOfPackageInfo	A list of complex types, each of which represents a deployment package .
ArrayOfString	A list of strings, each of which represents an identifier of an external list .
PackageInfo	Information about a deployment package.

3.1.4.2.3.1 ArrayOfString

Namespace: http://microsoft.com/webservices/SharePointPortalServer/ObaWS/

The **ArrayOfString** type represents a list of strings, each of which represents an identifier of an **external list**.

This complex type is defined as follows.

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

string: A string that represents an identifier of an external list of which the **deployment package** information will be returned. This identifier is a **GUIDString** or **curly braced GUID string**. This element **MUST** be presented.

3.1.4.2.3.2 ArrayOfPackageInfo

Namespace: http://microsoft.com/webservices/SharePointPortalServer/ObaWS/

The **ArrayOfPackageInfo** complex type is a list of complex types, each of which represents a **deployment package**. This complex type is defined as follows.

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

PackageInfo: A complex type that represents a deployment package.

3.1.4.2.3.3 PackageInfo

Namespace: http://microsoft.com/webservices/SharePointPortalServer/ObaWS/

The **PackageInfo** complex type that specifies the information about a **deployment package**. This complex type is defined as follows.

```
<xs:complexType name="PackageInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="PackageId" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="SelfSigned" type="xs:boolean"/>
  </xs:sequence>
```

</xs:complexType>

Url: The URL of the deployment package.

PackageId: The identifier of the deployment package. It is a **GUIDString**. When a deployment package is regenerated, a new package identifier is specified.

SelfSigned: true if the deployment package is signed by using a self-signed **certificate**.

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 UpsizeVirtualList

The **UpsizeVirtualList** operation generates a **Web Part Page** for viewing an item of an **external list** based on the specified **form template** and updates the viewing action of the external list to point to the Web Part Page.

The following is the **WSDL** port type specification of the **UpsizeVirtualList WSDL operation**.

```
<wsdl:operation name="UpsizeVirtualList" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:UpsizeVirtualListSoapIn"/>
  <wsdl:output message="tns:UpsizeVirtualListSoapOut"/>
</wsdl:operation>
```

The form template MUST conform to the form template format specified in [\[MS-IPFF2\]](#). If the **external content type** has a **Creator**, this operation also generates a Web Part Page for creating an item for the external list based on the same form template, and updates the creating action of the external list to point to the Web Part Page. If the external content type has an **Updater**, this operation also generates a Web Part Page for updating an item of the external list based on the same form template, and updates the updating action of the external list to point to the Web Part Page.

The protocol client sends an **UpsizeVirtualListSoapIn** request **WSDL message**, and the protocol server SHOULD [<5>](#) respond with an **UpsizeVirtualListSoapOut** response WSDL message, as follows:

1. The protocol client MUST specify the location of a form template that the protocol server maintains. The form template MUST conform to the form template format specified in [\[MS-IPFF2\]](#) and the location MUST be relative to the **site** that is associated with the **Web service**. If the location of the form template is not specified or the specified form template does not exist, the protocol server MUST return a **SOAP fault** as specified in [\[SOAP1.1\]](#) section 4.4 with fault code value of Server as specified in [\[SOAP1.1\]](#) section 4.4.1.

2. The protocol client **MUST** specify the namespace and name of the external content type on which the external list is based. If the namespace or name is not specified or is empty, the protocol server **MUST** return a SOAP fault as specified in [SOAP1.1] section 4.4 and the SOAP fault **MUST** include a fault code value of Server as specified in [SOAP1.1] section 4.4.1
3. There **MUST** be one external list associated with the parent folder of the folder in which the form template is contained. If the external list does not exist, the protocol server **MUST** return a SOAP fault as specified in [SOAP1.1] section 4.4 and the SOAP fault **MUST** include a fault code value of Server as specified in [SOAP1.1] section 4.4.1. This external list **MUST** be the one that the protocol server modifies.
4. The protocol server **MUST** generate a Web Part Page for viewing an item of the external list based on the specified form template and modify the viewing action to point to the page. If the page exists, the protocol server **MUST** overwrite the page.
5. If the external content type associated with the external list has a Creator, the protocol server **MUST** generate a Web Part Page for creating an item for the external list based on the specified form template, and modify the creating action to point to the page. If the page exists, the protocol server **MUST** overwrite the page.
6. If the external content type associated with the external list has an Updater, the protocol server **MUST** generate a Web Part Page for updating an item of the external list based on the specified form template, and modify the updating action to point to the page. If the page exists, the protocol server **MUST** overwrite the page.
7. If the user performing the client protocol does not have the permission to modify the external list, the protocol server **MUST** return a SOAP fault as specified in [SOAP1.1] section 4.4 and the SOAP fault **MUST** include a fault code value of Server as specified in [SOAP1.1] section 4.4.1.
8. If the user performing the client protocol does not have read permission on the site or the external list, the server **MUST** return **HTTP** status code 401 as specified in section 10 of [RFC2616].
9. If the user performing the client protocol has read permission on the site and the external list, but does not have the permission to create a page on the site associated with the external list the protocol server **MUST** return a SOAP fault as specified in [SOAP1.1] section 4.4 and the SOAP fault **MUST** include a fault code value of "tns:Server", where tns is defined in section 2.2.1.

3.1.4.3.1 Messages

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

Message	Description
UpsizeVirtualListSoapIn	The request WSDL message for the UpsizeVirtualList WSDL operation .
UpsizeVirtualListSoapOut	The response WSDL message for the UpsizeVirtualList WSDL operation .

3.1.4.3.1.1 UpsizeVirtualListSoapIn

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

The **SOAP action** value is:

```
http://microsoft.com/webservices/SharePointPortalServer/ObaWS/UpsizeVirtualList
```

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

3.1.4.3.1.2 UpsizeVirtualListSoapOut

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

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

3.1.4.3.2 Elements

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

Element	Description
UpsizeVirtualList	The input data for the UpsizeVirtualList WSDL operation .
UpsizeVirtualListResponse	The result data for the UpsizeVirtualList WSDL operation .

3.1.4.3.2.1 UpsizeVirtualList

The **UpsizeVirtualList** element specifies the input data for the **UpsizeVirtualList WSDL operation**.

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

xsnLocation: The relative URL of a **form template** that will be associated with the viewing, creating, and updating actions of an **external list**.

entityNamespace: The namespace of the **external content type** that is associated with the external list.

entityName: The name of the external content type that is associated with the external list.

3.1.4.3.2.2 UpsizeVirtualListResponse

The **UpsizeVirtualListResponse** element specifies the result data for the **UpsizeVirtualList WSDL operation**.

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

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.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples show compliant **SOAP** request and response messages in simple hypothetical scenarios. As noted in section [3.1](#), all operations are stateless, and each operation can be used independently of others.

4.1 Retrieve a List of Solution Package Locations Associated with a Set of External Lists

Client applications can retrieve the information about each of the **deployment packages** that are associated with a set of **external lists**. **Solution package** information is retrieved by using the **GetPackages** operation.

The following example shows the retrieval of the deployment package information.

Request

```
<?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>
    <GetPackages xmlns="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/">
      <listIds>
        <string>{6ce18be2-b829-46be-8b2e-1ec5d506a1b1}</string>
      </listIds>
      <clientVersion>15.0.0.0</clientVersion>
    </GetPackages>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?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>
    <GetPackagesResponse
xmlns="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/">
      <GetPackagesResult>
        <PackageInfo>
          <Url>http://example.com/Lists/Custom/ClientSolution/15/6ce18be2-b829-46be-8b2e-1ec5d506a1b1.vsto</Url>
          <PackageId>b878f7ab-d9b8-4592-80f7-f251a625aefa</PackageId>
          <SelfSigned>>true</SelfSigned>
        </PackageInfo>
      </GetPackagesResult>
    </GetPackagesResponse>
  </soap:Body>
</soap:Envelope>
```

4.2 Create a Profile Page for an Existing External Content Type

Client applications can create **profile pages** that are associated with **external content types**. Profile pages are generated by using the **CreateProfilePage** operation.

The following example shows the sequence that is used to create a profile page.

Request

```
<?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>
    <CreateProfilePage
xmlns="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/">
      <entityNamespace>CustomerNamespace</entityNamespace>
      <entityName>Customer</entityName>
      <allowOverwrite>true</allowOverwrite>
    </CreateProfilePage>
  </soap:Body>
</soap:Envelope>
```

Response

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

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided in this appendix.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation>Office Business Application Client Extensions Publish Web
Service</wsdl:documentation>
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/">
      <xs:element name="GetPackages">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="listIds" type="tns:ArrayOfString"/>
            <xs:element minOccurs="0" maxOccurs="1" name="clientVersion" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="ArrayOfString">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="GetPackagesResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetPackagesResult"
type="tns:ArrayOfPackageInfo"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="ArrayOfPackageInfo">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="PackageInfo" nillable="true"
type="tns:PackageInfo"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="PackageInfo">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Url" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="PackageId" type="xs:string"/>
          <xs:element minOccurs="1" maxOccurs="1" name="SelfSigned" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="CreateProfilePage">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="entityNamespace" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="entityName" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="allowOverwrite" type="xs:boolean"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreateProfilePageResponse">
        <xs:complexType/>
      </xs:element>
      <xs:element name="UpsizeVirtualList">
        <xs:complexType>
          <xs:sequence>
```

```

        <xs:element minOccurs="0" maxOccurs="1" name="xsnLocation" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="entityNamespace" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="entityName" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="UpsizeVirtualListResponse">
    <xs:complexType/>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:portType name="PublishServiceSoap">
    <wsdl:operation name="GetPackages">
        <wsdl:input message="tns:GetPackagesSoapIn"/>
        <wsdl:output message="tns:GetPackagesSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="CreateProfilePage">
        <wsdl:input message="tns:CreateProfilePageSoapIn"/>
        <wsdl:output message="tns:CreateProfilePageSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="UpsizeVirtualList">
        <wsdl:input message="tns:UpsizeVirtualListSoapIn"/>
        <wsdl:output message="tns:UpsizeVirtualListSoapOut"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PublishServiceSoap" type="tns:PublishServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetPackages">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/GetPackages"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateProfilePage">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/CreateProfilePage"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpsizeVirtualList">
        <soap:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/UpsizeVirtualList"
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="PublishServiceSoap12" type="tns:PublishServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetPackages">
        <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/GetPackages"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
    </wsdl:operation>

```

```

    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateProfilePage">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/CreateProfilePage"
style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="UpsizeVirtualList">
    <soap12:operation
soapAction="http://microsoft.com/webservices/SharePointPortalServer/ObaWS/UpsizeVirtualList"
style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:message name="CreateProfilePageSoapIn">
  <wsdl:part name="parameters" element="tns:CreateProfilePage"/>
</wsdl:message>
<wsdl:message name="CreateProfilePageSoapOut">
  <wsdl:part name="parameters" element="tns:CreateProfilePageResponse"/>
</wsdl:message>
<wsdl:message name="GetPackagesSoapIn">
  <wsdl:part name="parameters" element="tns:GetPackages"/>
</wsdl:message>
<wsdl:message name="GetPackagesSoapOut">
  <wsdl:part name="parameters" element="tns:GetPackagesResponse"/>
</wsdl:message>
<wsdl:message name="UpsizeVirtualListSoapIn">
  <wsdl:part name="parameters" element="tns:UpsizeVirtualList"/>
</wsdl:message>
<wsdl:message name="UpsizeVirtualListSoapOut">
  <wsdl:part name="parameters" element="tns:UpsizeVirtualListResponse"/>
</wsdl:message>
</wsdl:definitions>

```

7 Appendix B: Product Behavior

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

- Microsoft FAST Search Server 2010
- Microsoft FAST Search Server 2010 for SharePoint
- Microsoft FAST Search Server 2010 for SharePoint Internet Sites
- Microsoft Office 2010 suites
- Microsoft SharePoint Server 2010 Enterprise
- Microsoft SharePoint 2010 for Internet Sites Standard
- Microsoft SharePoint 2010 for Internet Sites Enterprise
- Microsoft Search Server 2010
- Microsoft Search Server 2010 Express
- Microsoft SharePoint Designer 2010
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Workspace 2010
- Microsoft Office 2013
- Microsoft SharePoint Designer 2013
- Microsoft SharePoint Server 2013
- Microsoft Office 2016
- Microsoft SharePoint Server 2016
- Microsoft Office 2019
- Microsoft SharePoint Server 2019
- Microsoft Office 2021
- Microsoft SharePoint Server Subscription Edition
- Microsoft Office 2024 Preview

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

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

[<1> Section 3.1.4.1](#): Search Server 2010 Express, Search Server 2010, FAST Search Server 2010 for SharePoint Internet Sites, FAST Search Server 2010 for SharePoint, SharePoint 2010 FIS Enterprise, SharePoint 2010 FIS Standard, and SharePoint Server 2010 also create or update an **Action** with the

name "View Profile" and set the **Action** as the default **Action** of the specified **external content type** as follows. If the external content type does not have an **Action** with the name "View Profile", the server creates one. The URL of the **Action** is set to specify the **profile page** and for each **Identifier** of the external content type an **ActionParameter** is created with the same name as the **Identifier**. In case the external content type has an **Action** with the name "View Profile", if the **Action** has a property **Office Version** and the property value is *14*, the **Action** will be reused. All the **ActionParameters** of the **Action** are removed and a new **ActionParameter** is created with the same name as the **Identifier** for each **Identifier**. If the property **Office Version** does not exist or the property value is not *14*, the **Action** is renamed to "View Profile (SharePoint 2007)". A new **Action** with the name "View Profile" is created, and for each **Identifier** of the external content type an **ActionParameter** is created with the same name as the **Identifier**. In both cases, the URL of the "View Profile" **Action** is set to specify the profile page.

<2> [Section 3.1.4.1](#): Search Server 2010 Express, Search Server 2010, FAST Search Server 2010 for SharePoint Internet Sites, FAST Search Server 2010 for SharePoint, SharePoint 2010 FIS Enterprise, SharePoint 2010 FIS Standard and SharePoint Server 2010 construct a profile page URL as follows. The server first checks the default **Action** of the external content type. If the default **Action** name is "View Profile" and the **Action** URL refers to a file in the folder *_bdc/<encoded-namespace>* of the site that hosts profile pages, the server constructs the profile page URL by using the schema, authority, and path of the **Action** URL. The *encoded-namespace* is a string generated from the namespace of the external content type as follows: first replace with *'_'* the invalid characters (*'~', '#', '%', '&', '{, }'* and *'.'*), then percent-encode the result and finally truncate the encoded string to at most 128 characters without breaking in the middle of a percent-encoding triplet, for example, *"%25"*. If a file with the same URL exists, the file will be overwritten. If the default **Action** name is not "View Profile", or the **Action** URL refers to a file that is not in the folder as specified in this paragraph, the server generates the profile page URL as *<site-url>/_bdc/<encoded-namespace>/<encoded-name>_<version-tag>.aspx*, where *site-url* is the URL of the site hosting profile pages, *encoded-name* is a string generated from the external content type name in the same way as *encoded-namespace*, except that the encoded string is truncated to no more than 124 characters, and **version** tag is an integer. The profile page URL is first constructed by using *1* as the **version** tag. If a file exists with the same URL, the server keeps constructing a new URL by increasing the **version** tag until either no file with the same URL exists or the **version** tag is *999*. In the latter case, the existing file will be overwritten.

<3> [Section 3.1.4.1](#): Search Server 2010 Express, Search Server 2010, FAST Search Server 2010 for SharePoint Internet Sites, FAST Search Server 2010 for SharePoint, SharePoint 2010 FIS Enterprise, SharePoint 2010 FIS Standard and SharePoint Server 2010 return a **SOAP fault** if the **site** that hosts profile pages is not in the local **farm**.

<4> [Section 3.1.4.2](#): SharePoint Server 2010 Enterprise and SharePoint 2010 FIS Enterprise respond with a **GetPackagesSoapOut** response **WSDL message**. Any other Microsoft implementation returns a SOAP fault that includes a fault-code value of *Server* as specified in [\[SOAP1.1\]](#) section 4.4.1.

<5> [Section 3.1.4.3](#): SharePoint Server 2010 Enterprise server and SharePoint 2010 FIS Enterprise server respond with an **UpsizeVirtualListSoapOut** response WSDL message. Any other Microsoft implementation returns a SOAP fault that includes a fault-code value of *Server* as specified in [\[SOAP1.1\]](#) section 4.4.1.

8 Change Tracking

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

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

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

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

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

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

Section	Description	Revision class
Z Appendix B: Product Behavior	Updated list of supported products.	Major

9 Index

A

Abstract data model
[server](#) 14
[Applicability](#) 11
[Attribute groups](#) 13
[Attributes](#) 13

C

[Capability negotiation](#) 11
[Change tracking](#) 34
[Common data structures](#) 13
[Complex types](#) 13
[Create a profile page for an existing external content type example](#) 26

D

Data model - abstract
[server](#) 14

E

Events
[local - server](#) 25
[timer - server](#) 25
Examples
[create a profile page for an existing external content type](#) 26
[overview](#) 26
[retrieve a list of solution package locations associated with a set of external lists](#) 26

F

[Fields - vendor-extensible](#) 11
[Full WSDL](#) 29

G

GetPackages operation ([section 3.1.4](#) 15, [section 3.1.4.2](#) 18)
[Glossary](#) 6
[Groups](#) 13

I

[Implementer - security considerations](#) 28
[Index of security parameters](#) 28
[Informative references](#) 9
Initialization
[server](#) 15
[Introduction](#) 6

L

Local events
[server](#) 25

M

Message processing
[server](#) 15
Messages
[attribute groups](#) 13
[attributes](#) 13
[common data structures](#) 13
[complex types](#) 13
[elements](#) 12
[enumerated](#) 12
[groups](#) 13
[namespaces](#) 12
[simple types](#) 13
[syntax](#) 12
[transport](#) 12

N

[Namespaces](#) 12
[Normative references](#) 9

O

Operations
[CreateProfilePage](#) 15
[GetPackages](#) 18
[UpsizeVirtualList](#) 22
[Overview \(synopsis\)](#) 10

P

[Parameters - security index](#) 28
[Preconditions](#) 10
[Prerequisites](#) 10
[Product behavior](#) 32
Protocol Details
[overview](#) 14

R

[References](#) 9
[informative](#) 9
[normative](#) 9
[Relationship to other protocols](#) 10
[Retrieve a list of solution package locations associated with a set of external lists example](#) 26

S

Security
[implementer considerations](#) 28
[parameter index](#) 28
Sequencing rules
[server](#) 15
Server
[abstract data model](#) 14
[CreateProfilePage operation](#) 15
[GetPackages operation](#) 18
[initialization](#) 15

[local events](#) 25
[message processing](#) 15
[sequencing rules](#) 15
[timer events](#) 25
[timers](#) 15
[UpsizeVirtualList operation](#) 22
[Simple types](#) 13
[Standards assignments](#) 11
Syntax
[messages - overview](#) 12

T

Timer events
[server](#) 25
Timers
[server](#) 15
[Tracking changes](#) 34
[Transport](#) 12
Types
[complex](#) 13
[simple](#) 13

U

[UpsizeVirtualList operation](#) ([section 3.1.4](#) 15, [section 3.1.4.3](#) 22)

V

[Vendor-extensible fields](#) 11
[Versioning](#) 11

W

[WSDL](#) 29