

[MS-BCSDWPS]: Business Connectivity Services Deployment Web Service Protocol Specification

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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). 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.
- **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
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Major	Updated and revised the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
2 Messages	9
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	10
2.2.4 Complex Types	10
2.2.5 Simple Types	10
2.2.6 Attributes	10
2.2.7 Groups	10
2.2.8 Attribute Groups	10
3 Protocol Details	11
3.1 Server Details	11
3.1.1 Abstract Data Model	11
3.1.2 Timers	12
3.1.3 Initialization	12
3.1.4 Message Processing Events and Sequencing Rules	12
3.1.4.1 CreateProfilePage	13
3.1.4.1.1 Messages	14
3.1.4.1.1.1 CreateProfilePageSoapIn	14
3.1.4.1.1.2 CreateProfilePageSoapOut	14
3.1.4.1.2 Elements	14
3.1.4.1.2.1 CreateProfilePage	14
3.1.4.1.2.2 CreateProfilePageResponse	15
3.1.4.1.3 Complex Types	15
3.1.4.1.4 Simple Types	15
3.1.4.1.5 Attributes	15
3.1.4.1.6 Groups	15
3.1.4.1.7 Attribute Groups	15
3.1.4.2 GetPackages	15
3.1.4.2.1 Messages	16
3.1.4.2.1.1 GetPackagesSoapIn	16
3.1.4.2.1.2 GetPackagesSoapOut	17
3.1.4.2.2 Elements	17
3.1.4.2.2.1 GetPackages	17
3.1.4.2.2.2 GetPackagesResponse	17

3.1.4.2.3	Complex Types	17
3.1.4.2.3.1	ArrayOfString.....	18
3.1.4.2.3.2	ArrayOfPackageInfo	18
3.1.4.2.3.3	PackageInfo	18
3.1.4.2.4	Simple Types.....	19
3.1.4.2.5	Attributes.....	19
3.1.4.2.6	Groups.....	19
3.1.4.2.7	Attribute Groups	19
3.1.4.3	UpsizeVirtualList.....	19
3.1.4.3.1	Messages	20
3.1.4.3.1.1	UpsizeVirtualListSoapIn	20
3.1.4.3.1.2	UpsizeVirtualListSoapOut.....	20
3.1.4.3.2	Elements.....	21
3.1.4.3.2.1	UpsizeVirtualList.....	21
3.1.4.3.2.2	UpsizeVirtualListResponse	21
3.1.4.3.3	Complex Types	21
3.1.4.3.4	Simple Types.....	21
3.1.4.3.5	Attributes.....	21
3.1.4.3.6	Groups.....	21
3.1.4.3.7	Attribute Groups	21
3.1.5	Timer Events	22
3.1.6	Other Local Events	22
4	Protocol Examples.....	23
4.1	Retrieve a List of Solution Package Locations Associated with a Set of External Lists	23
4.2	Create a Profile Page for an Existing External Content Type	23
4.3	Associate a Form with an External List.....	24
5	Security.....	26
5.1	Security Considerations for Implementers.....	26
5.2	Index of Security Parameters	26
6	Appendix A: Full WSDL.....	27
7	Appendix B: Product Behavior.....	30
8	Change Tracking.....	32
9	Index	33

1 Introduction

This document specifies the Business Connectivity Services Deployment Web Service Protocol. This 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.

1.1 Glossary

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

- certificate**
- curly braced GUID string**
- globally unique identifier (GUID)**
- GUIDString**
- Hypertext Transfer Protocol (HTTP)**
- Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**
- Transmission Control Protocol (TCP)**

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

- Action**
- ActionParameter**
- Business Connectivity Services (BCS)**
- Creator**
- deployment package**
- external content type**
- external list**
- farm**
- file**
- folder**
- form**
- form template**
- form template (.xsn) file**
- Identifier**
- LobSystem**
- MetadataObject**
- profile page**
- site**
- SOAP (Simple Object Access Protocol)**
- SOAP action**
- SOAP body**
- SOAP fault**
- solution package**
- Updater**
- view**
- Web Part Page**
- Web service**
- WSDL (Web Services Description Language)**
- WSDL message**
- WSDL operation**
- XML Schema**

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

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

[MS-BCSDPFFS] Microsoft Corporation, "[Business Connectivity Services Deployment Package File Format Specification](#)", July 2009.

[MS-IPFF2] Microsoft Corporation, "[InfoPath Form Template Format Version 2](#)", July 2009.

[MS-OSCO] Microsoft Corporation, "[Office Server ClickOnce Manifest Structure Specification](#)", July 2009.

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

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

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

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

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

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

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

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

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

1.2.2 Informative References

[IANAPORT] Internet Assigned Numbers Authority, "Port Numbers", November 2006, <http://www.iana.org/assignments/port-numbers>

[MC-NMF] Microsoft Corporation, "[.NET Message Framing Protocol Specification](#)", October 2007.

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

[MS-OFGLS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)", June 2008.

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

1.3 Protocol Overview (Synopsis)

This protocol 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 [\[SOAP1.1\]](#), [\[SOAP1.2/1\]](#) and [\[SOAP1.2/2\]](#). It transmits those messages by using **HTTP**, as described in [\[RFC2616\]](#), **HTTPS**, as described in [\[RFC2818\]](#), or **TCP**, as described in [\[MC-NMF\]](#).

The Business Connectivity Services Deployment Web Service Protocol uses SOAP over HTTP or HTTPS as shown in the following layering diagram:

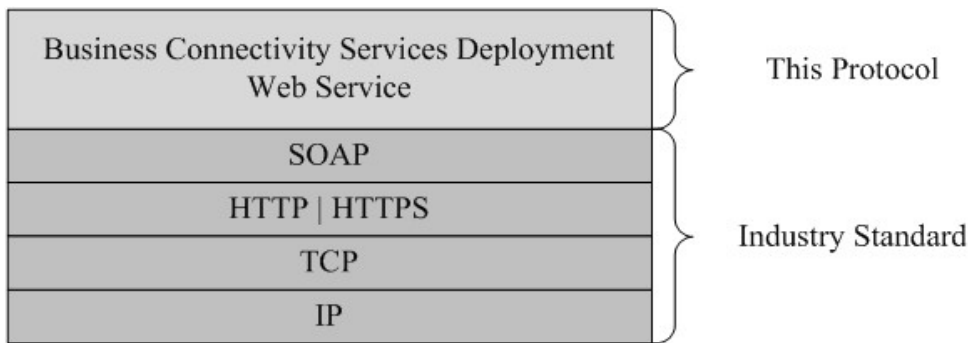


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site (2)** that is identified by a URL that protocol clients recognize. The protocol server endpoint is formed by appending "/_vti_bin/PublishService.asmx" to the URL of the site (2)—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 specified 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 be less restrictive than 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 specifies additional restrictions 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, present, and not null.

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 the common definitions that are used by this protocol. The syntax of the definitions uses the **XML Schema** as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language as defined 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
(none)	http://microsoft.com/webservices/SharePointPortalServer/ObaWS/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
env	http://schemas.xmlsoap.org/soap/envelope/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://microsoft.com/webservices/SharePointPortalServer/ObaWS/	
wsdl	http://schemas.xmlsoap.org/wsdl	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

None.

2.2.3 Elements

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

2.2.4 Complex Types

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

In the following sections, the schema definition might be less restrictive than 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 specifies additional restrictions 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, present, and not null.

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 (2) 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 (2) that hosts profile pages. All profile pages will be generated on this site (2)

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 **globally unique identifier (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

This specification includes the following **WSDL operations**:

WSDL 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 (2) that are associated with a list of external lists. If the deployment package (2) of an external list does not exist or the external list has been modified since the deployment package (2) was generated, this operation MUST also generate a new deployment package (2) for the external list. The deployment packages (2) MUST conform to the format specified in both [MS-BCSDPFFS] and [MS-OSCO] .
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

WSDL Operation	Description
	of the external list to point to the Web Part Page. 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.

3.1.4.1 CreateProfilePage

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

```
<wsdl:operation name="CreateProfilePage">
  <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 of which the profile page will be created.
- The protocol client MUST specify whether the profile page to be created overwrites any previously created profile page.

The protocol server MUST [<1>](#) generate a profile page for the particular external content type with an implementation specific URL [<2>](#).

In certain cases, the protocol server MUST return a SOAP fault as specified in [\[SOAP1.1\]](#) section 4.4. The SOAP fault MUST include a faultcode value of "env:Server", where env is defined in section [2.2.1](#). 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 (2) that hosts profile pages is not specified.
- The site (2) that hosts profile pages does not exist [<3>](#).
- The site (2) 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 (2) that hosts profile pages.

3.1.4.1.1 Messages

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

The SOAP action value is:

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

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

3.1.4.1.2 Elements

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

3.1.4.1.2.1 CreateProfilePage

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

```
<xs:element name="CreateProfilePage">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="entityNamespace" type="xs:string"/>
      <xs:element minOccurs="1" 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". The value of this element MUST be ignored by the protocol server.

3.1.4.1.2.2 CreateProfilePageResponse

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

```
<xs:element name="CreateProfilePageResponse">
  <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 (2) that are associated with a list of external lists. If the deployment package (2) of an external list does not exist or the external list has been modified since the deployment package (2) was generated, this operation **MUST** also generate a new deployment package (2) for the external list. The deployment packages (2) **MUST** conform to the format specified in both [\[MS-BCSDPFFS\]](#) and [\[MS-OSCO\]](#).

```
<wsdl:operation name="GetPackages">
  <wsdl:input message="tns:GetPackagesSoapIn"/>
  <wsdl:output message="tns:GetPackagesSoapOut"/>
</wsdl:operation>
```

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 of which the deployment package (2) information will be returned.
- If the protocol server determines that the deployment package (2) of an external list does not exist, the protocol server **MUST** generate a deployment package (2) for the external list.
- If there is a deployment package (2) for an external list, the protocol server **MUST** check whether changes were made to the external list after the deployment package (2) was generated. When

any of the following changes occur, the protocol server MUST generate a new deployment package (2):

- 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 (2) that correspond to the external lists.

In certain cases, the protocol server MUST return a SOAP fault as specified in [\[SOAP1.1\]](#) section 4.4. The SOAP fault MUST include a faultcode value of "env:Server", where env is defined in section [2.2.1](#). 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 **GUIDString** or **curly braced GUID string**.
- The external list that is specified by any identifier does not exist on the site (2) that is associated with the Web service.
- The site (2) 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 (2) or the external lists, the server MUST return HTTP status code 401 as specified in section 10 of [\[RFC2616\]](#).

3.1.4.2.1 Messages

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

The SOAP action value is:

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

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

3.1.4.2.2 Elements

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

3.1.4.2.2.1 GetPackages

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

```
<xs:element name="GetPackages">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="listIds" type="tns:ArrayOfString"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

listIds: The list of external list identifiers of which the deployment package (2) information will be returned. Each identifier is a GUIDString or curly braced GUID string. This element **MUST** be presented.

3.1.4.2.2.2 GetPackagesResponse

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

```
<xs:element name="GetPackagesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="GetPackagesResult"
type="tns:ArrayOfPackageInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetPackagesResult: An array of complex type PackageInfo that represents deployment packages (2).

3.1.4.2.3 Complex Types

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

3.1.4.2.3.1 ArrayOfString

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">
  <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 (2) 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

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">
  <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

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

```
<xs:complexType name="PackageInfo">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Url" type="xs:string"/>
    <xs:element minOccurs="1" 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 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.

```
<wsdl:operation name="UpsizeVirtualList">
  <wsdl:input message="tns:UpsizeVirtualListSoapIn"/>
  <wsdl:output message="tns:UpsizeVirtualListSoapOut"/>
</wsdl:operation>
```

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

- 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 (2) 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 faultcode="env:Server", where env is defined in section [2.2.1](#).
- 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 empty, the protocol server **MUST** return a SOAP fault as specified in [\[SOAP1.1\]](#) section 4.4 and the SOAP fault **MUST** include a faultcode value of "env:Server", where env is defined in section [2.2.1](#).
- 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 faultcode value of "env:Server", where env is defined in section [2.2.1](#). This external list **MUST** be the one that the protocol server modifies.

- 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.
- 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.
- 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.
- 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 faultcode value of "env:Server", where tns is defined in section [2.2.1](#).
- If the user performing the client protocol does not have read permission on the site (2) or the external list, the server MUST return HTTP status code 401 as specified in section 10 of [\[RFC2616\]](#).
- If the user performing the client protocol has read permission on the site (2) and the external list, but does not have the permission to create a page on the site (2) 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 faultcode value of "tns:Server", where tns is defined in section [2.2.1](#).

3.1.4.3.1 Messages

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

The SOAP action value is:

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

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

3.1.4.3.2 Elements

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

3.1.4.3.2.1 UpsizeVirtualList

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

```
<xs:element name="UpsizeVirtualList">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="xsnLocation" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="entityNamespace" type="xs:string"/>
      <xs:element minOccurs="1" 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 result data for the **UpsizeVirtualList** WSDL operation.

```
<xs:element name="UpsizeVirtualListResponse">
  <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-1ec5d506alb1}</string>
      </listIds>
    </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/6ce18be2-b829-46be-8b2e-1ec5d506alb1.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>
```

4.3 Associate a Form with an External List

Client applications can modify the create, view, and update actions of the external list to point to a **form template (.xsn) file** at a known location on the protocol server. The association between an external list and a form template (.xsn) file can be created using the **UpsizeVirtualList** operation.

The following example shows the sequence that is used to associate a form template (.xsn) file with an external list.

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 below.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions 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 xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">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="1" maxOccurs="1" name="listIds" type="tns:ArrayOfString"
/>
          </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="1" 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="1" maxOccurs="1" name="Url" type="xs:string" />
          <xs:element minOccurs="1" 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="1" maxOccurs="1" name="entityNamespace" type="xs:string"
/>
            <xs:element minOccurs="1" 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:schema>
  </wsdl:types>

```

```

        </xs:complexType>
    </xs:element>
    <xs:element name="CreateProfilePageResponse">
        <xs:complexType />
    </xs:element>
    <xs:element name="UpsizeVirtualList">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="1" maxOccurs="1" name="xsnLocation" type="xs:string" />
                <xs:element minOccurs="1" maxOccurs="1" name="entityNamespace" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="UpsizeVirtualListResponse">
        <xs:complexType />
    </xs:element>
</xs:schema>
</wsdl:types>
<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="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="UpsizeVirtualListSoapIn">
    <wsdl:part name="parameters" element="tns:UpsizeVirtualList" />
</wsdl:message>
<wsdl:message name="UpsizeVirtualListSoapOut">
    <wsdl:part name="parameters" element="tns:UpsizeVirtualListResponse" />
</wsdl:message>
<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:definitions>

```

7 Appendix B: Product Behavior

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

- Microsoft® FAST™ Search Server 2010
- Microsoft® Office 2010 suites
- Microsoft® SharePoint® Designer 2010
- Microsoft® SharePoint® Server 2010
- Microsoft® SharePoint® Workspace 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.

[<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 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 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 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 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 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, e.g. *"%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 as above, 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, *version-tag* is an integer, and *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. The profile page URL is first constructed by using 1 as *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 (2) 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 faultcode value of "env:Server", where env is "http://schemas.xmlsoap.org/soap/envelope/".

<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 faultcode value of "env:Server", where env is defined in section [2.2.1](#).

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

A

Abstract data model
[server](#) 11
[Applicability](#) 8
[Associate a form with an external list example](#) 24
[Attribute groups](#) 10
[Attributes](#) 10

C

[Capability negotiation](#) 8
[Change tracking](#) 32
[Complex types](#) 10
[Create a profile page for an existing external content type example](#) 23

D

Data model - abstract
[server](#) 11

E

Events
[local - server](#) 22
[timer - server](#) 22
Examples
[associate a form with an external list](#) 24
[create a profile page for an existing external content type](#) 23
[overview](#) 23
[retrieve a list of solution package locations associated with a set of external lists](#) 23

F

[Fields - vendor-extensible](#) 8
[Full WSDL](#) 27

G

[Glossary](#) 5
[Groups](#) 10

I

[Implementer - security considerations](#) 26
[Index of security parameters](#) 26
[Informative references](#) 6
Initialization
[server](#) 12
[Introduction](#) 5

L

Local events
[server](#) 22

M

Message processing
[server](#) 12
Messages
[attribute groups](#) 10
[attributes](#) 10
[complex types](#) 10
[elements](#) 10
[enumerated](#) 9
[groups](#) 10
[namespaces](#) 9
[simple types](#) 10
[syntax](#) 9
[transport](#) 9

N

[Namespaces](#) 9
[Normative references](#) 6

O

Operations
[CreateProfilePage](#) 13
[GetPackages](#) 15
[UpsizeVirtualList](#) 19
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 26
[Preconditions](#) 8
[Prerequisites](#) 8
[Product behavior](#) 30

R

References
[informative](#) 6
[normative](#) 6
[Relationship to other protocols](#) 7
[Retrieve a list of solution package locations associated with a set of external lists example](#) 23

S

Security
[implementer considerations](#) 26
[parameter index](#) 26
Sequencing rules
[server](#) 12
Server
[abstract data model](#) 11
[CreateProfilePage operation](#) 13
[GetPackages operation](#) 15
[initialization](#) 12
[local events](#) 22

[message processing](#) 12
[overview](#) 11
[sequencing rules](#) 12
[timer events](#) 22
[timers](#) 12
[UpsizeVirtualList operation](#) 19
[Simple types](#) 10
[Standards assignments](#) 8
Syntax
[messages - overview](#) 9

T

Timer events
[server](#) 22
Timers
[server](#) 12
[Tracking changes](#) 32
[Transport](#) 9
Types
[complex](#) 10
[simple](#) 10

V

[Vendor-extensible fields](#) 8
[Versioning](#) 8

W

[WSDL](#) 27