[MS-CONATB]:
Content Area Toolbox Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

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

Date	Revision History	Revision Class	Comments
02/11/2013	2.6	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	2.7	Minor	Clarified the meaning of the technical content.
11/18/2013	2.7	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	2.7	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	2.8	Minor	Clarified the meaning of the technical content.
07/31/2014	2.8	No change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	2.8	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

The Content Area Toolbox Web Service Protocol can be used by protocol clients to enumerate the Web controls that are stored on a protocol server and can be used on a Web page.

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

1.1   Glossary

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

Augmented Backus-Naur Form (ABNF)
content type
content type identifier
culture name
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
page layout
site
SOAP
SOAP action
SOAP body
SOAP fault
SOAP message
Uniform Resource Locator (URL)
web control
Web Services Description Language (WSDL)
WSDL message
WSDL operation
XML document
XML namespace
XML namespace prefix
XML schema

The following terms are specific to this document:

localize: The process of adapting an application or documentation, including text and non-text elements, to meet the language, cultural, and political expectations and requirements of a specific geographic country or region.

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

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

1.2.1 Normative References

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

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

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

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

[RFC4646] Phillips, A., and Davis, M., Eds., "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006, http://www.rfc-editor.org/rfc/rfc4646.txt

[RFC4647] Phillips, A., and Davis, M., Eds., "Matching of Language Tags", BCP 47, RFC 4647, September 2006, http://www.rfc-editor.org/rfc/rfc4647.txt

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

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

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

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

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

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

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

1.2.2 Informative References

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

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

1.3 Overview

This protocol allows a protocol client to enumerate the **Web controls** that are stored on a protocol server and can be used on a Web page. A typical scenario for using this protocol is a Web-page-

editing application that queries the protocol server for available Web controls and displays those controls to a user, who can then select and insert a Web control into a Web page.

1.4 Relationship to Other Protocols

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

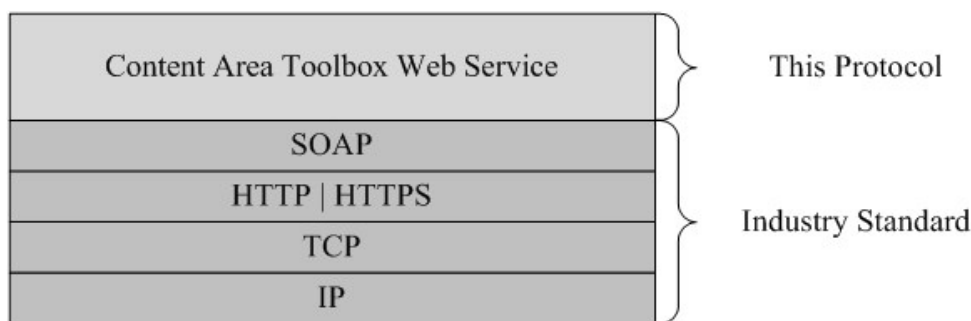The Content Area Toolbox Web Service Protocol uses SOAP over HTTP as shown in the following layering diagram:

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 is known by protocol clients. The protocol server endpoint is formed by appending "/_vti_bin/ContentAreaToolboxService.asmx" to the URL of the site (2), for example http://www.example.com/Repository/_vti_bin/ContentAreaToolboxService.asmx.

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

1.6 Applicability Statement

This protocol was designed for toolbox panels that contain fewer than or equal to 100 Web controls.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

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

- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section 3.1.4.

1.8 Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS to help secure communications with protocol clients.

Protocol messages MUST be formatted as specified in [SOAP1.1] section 4 or [SOAP1.2/1] section 5. Protocol server faults MUST be returned by using either HTTP status codes, as specified in [RFC2616] section 10, or **SOAP faults**, as specified in [SOAP1.1] section 4.4 or [SOAP1.2/1] section 5.4.

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

2.2.4 Complex Types

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

2.2.5   Simple Types

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

2.2.6   Attributes

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

2.2.7   Groups

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

2.2.8   Attribute Groups

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

3   Protocol Details

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

Except where specified, protocol clients SHOULD interpret HTTP status codes that are returned by the protocol server as specified in [RFC2616] section 10.

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 can be interpreted by protocol clients in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults by using either HTTP status codes or SOAP faults, as specified in this section.

This protocol allows protocol servers to perform implementation-specific localization of text in various messages. Except where specified, the localization of this text is an implementation-specific behavior of the protocol server and is not significant for interoperability.

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 stores a collection of Web pages that are identified by URLs. Each Web page has an associated **content type**.

The protocol server also maintains a list of toolbox panels, which are mappings from a type of Web page, which is referred to as a content type, to the set of Web controls that can be used on that type of page.

The protocol server additionally maintains a shared toolbox panel, which contains a mapping of the set of Web controls that can be used by all types of Web pages.

Each toolbox panel has an associated identifier.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Message Processing Events and Sequencing Rules

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

Operation	Description
FetchControlLists	Enumerates the list of Web controls for a specific toolbox panel.
FetchPanelsInformationByUrl	Enumerates information about the toolbox panels that are available for a Web page.

3.1.4.1   FetchControlLists

The **FetchControlLists** operation enumerates the list of Web controls to display in a toolbox panel.

```
<wsdl:operation name="FetchControlLists">
    <wsdl:input message="tns:FetchControlListsSoapIn" />
    <wsdl:output message="tns:FetchControlListsSoapOut" />
</wsdl:operation>
```

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

If the toolbox panel with the identifier specified in the **controlListIds** element does not exist, then the protocol server MUST return to the protocol client a SOAP fault with **faultcode** (as specified in [SOAP1.1] section 4.4.1) set to Client. Otherwise, the protocol server MUST return information about the Web controls for the toolbox panel in the **FetchControlListsResult** element.

3.1.4.1.1   Messages

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

Message	Description
FetchControlListsSoapIn	The request **WSDL message** for a **FetchControlLists** WSDL operation.
FetchControlListsSoapOut	The response WSDL message for a **FetchControlLists** WSDL operation.

3.1.4.1.1.1   FetchControlListsSoapIn

The request WSDL message for a **FetchControlLists** WSDL operation (section 3.1.4.1).

The **SOAP action** value is:

```
http://schemas.microsoft.com/sharepoint/publishing/soap/FetchControlLists
```

The **SOAP body** contains a **FetchControlLists** element (section 3.1.4.1.2.1).

3.1.4.1.1.2   FetchControlListsSoapOut

The response WSDL message for a **FetchControlLists** WSDL operation (section 3.1.4.1).

The SOAP body contains a **FetchControlListsResponse** element (section 3.1.4.1.2.2).

3.1.4.1.2 Elements

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

Element	Description
FetchControlLists	The input data of a **FetchControlLists** WSDL operation.
FetchControlListsResponse	The result data of a **FetchControlLists** WSDL operation.

3.1.4.1.2.1 FetchControlLists

The input data of a **FetchControlLists** WSDL operation (section 3.1.4.1).

```
<s:element name="FetchControlLists">
  <s:complexType>
    <s:sequence>
      <s:element name="controlListIds"/>
        <s:sequence>
          <s:element name="string" type="s:string" maxOccurs="unbounded"/>
        </s:sequence>
      <s:element name="cultureName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

controlListIds: An array that contains a list of toolbox panel identifiers.

controlListIds.string: A **content type identifier**.

cultureName: The language tag, as specified in [RFC4646] and [RFC4647], that is used to **localize** data in the response WSDL message.

3.1.4.1.2.2 FetchControlListsResponse

The result data of a **FetchControlLists** WSDL operation (section 3.1.4.1).

```
<s:element name="FetchControlListsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="FetchControlListsResult">
        <s:complexType>
          <s:sequence>
            <s:element name="string" type="s:string" minOccurs="0" maxOccurs="unbounded"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
```

FetchControlListsResult: An array that contains a list of Web controls.
FetchControlListsResult.string: An **XML document** that MUST conform to the following XML schema:

```
<s:schema elementFormDefault="unqualified" >
<s:element name="ControlsList" type="ControlsListType"/>
</s:schema>
```

3.1.4.1.3 Complex Types

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

Complex type	Description
ControlsListType	Defines a set of Web controls for a toolbox panel.
AssemblyType	Defines a group of Web controls.
ControlType	Contains the metadata of a Web control.

3.1.4.1.3.1 ControlsListType

The **ControlsListType** complex type defines a set of Web controls for a toolbox panel.

```
<s:complexType name="ControlsListType">
  <s:sequence>
    <s:element name="Assembly" type="AssemblyType" minOccurs="0"
     maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

Assembly: A set of Web controls. The element is present when the result data of a **FetchControlLists** WSDL operation returns at least one web control for a toolbox panel.

3.1.4.1.3.2 AssemblyType

The **AssemblyType** complex type defines a group of Web controls.

```
<s:complexType name="AssemblyType">
  <s:sequence>
    <s:element name="Control" type="ControlType"
     maxOccurs="unbounded"/>
  </s:sequence>
  <s:attribute name="Name" type="s:string" use="required"/>
  <s:attribute name="Namespace" type="s:string" use="required"/>
  <s:attribute name="TagPrefix" type="s:string" use="required"/>
</s:complexType>
```

Control: The set of metadata for the Web control, as specified in section 3.1.4.1.3.3.

Name: The category for the group of Web controls.

Namespace: The subcategory for the group of Web controls.

TagPrefix: The identifier the group of Web controls.

3.1.4.1.3.3 ControlType

The **ControlType** complex type defines the metadata of a Web control.

```
<s:complexType name="ControlType">
  <s:attribute name="Name" type="s:string" use="required"/>
  <s:attribute name="Title" type="s:string" use="required"/>
  <s:attribute name="Description" type="s:string" use="required"/>
  <s:attribute name="Template" type="s:string" use="required"/>
</s:complexType >
```

Name: Reserved. This value MUST be the same as the **Title** attribute and MUST be ignored by the protocol client.

Title: The display name of the Web control.

Description: The description of the Web control.

Template: The HTML snippet for the Web control. The protocol server generates this string by replacing "<tag-prefix>" with "{0}" in the following **Augmented Backus-Naur Form (ABNF)** grammar.

```
<template> = '<' tag-prefix ':' <control-name> <control-properties> '> </' <tag-prefix> ':'
<control-name> '>'
<control-name> = 1*(TEXT)
<tag-prefix> = 1*(ALPHA)
<control-properties> = *<control-property>
<control-property> = ' ' <property-name> '=' DQUOTE <property-value> DQUOTE
<property-name> = 1*(ALPHA)
<property-value> = 1*(qdtext)
DQUOTE = <US-ASCII double-quote mark (34)>
qdtext = *QSAFE-CHAR
```

An example HTML snippet for the **DateTime** Web control:

```
<{0}:DateTime runat="server"> </{0}:DateTime>
```

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   FetchPanelsInformationByUrl

The **FetchPanelsInformationByUrl** operation enumerates information about the toolbox panels that are available for a Web page, based on the content type that is associated with the page.

```
<wsdl:operation name="FetchPanelsInformationByUrl">
    <wsdl:input message="tns:FetchPanelsInformationByUrlSoapIn" />
    <wsdl:output message="tns:FetchPanelsInformationByUrlSoapOut" />
</wsdl:operation>
```

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

If the Web page identified by the **pageLayoutURL** element does not exist in the collection of Web pages, then the protocol server MUST return to the protocol client a SOAP fault with **faultcode** set to Server.

Next, if the content type that is associated with the Web page does not exist in the list of toolbox panels, then the protocol server MUST return to the protocol client a SOAP fault with **faultcode** set to Server.

Otherwise, the protocol server MUST return a **FetchPanelsInformationByUrlResult** element that contains two **PanelInfo** elements. One **PanelInfo** element contains information about the shared toolbox panel, which is available to all types of Web pages. The other **PanelInfo** element contains information about the toolbox panel that maps to the content type that is associated with the Web page.

3.1.4.2.1   Messages

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

Message	Description
FetchPanelsInformationByUrlSoapIn	The request WSDL message for a **FetchPanelsInformationByUrl** WSDL operation.
FetchPanelsInformationByUrlSoapOut	The response WSDL message for a **FetchPanelsInformationByUrl** WSDL operation.

3.1.4.2.1.1   FetchPanelsInformationByUrlSoapIn

The request WSDL message for a **FetchPanelsInformationByUrl** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/sharepoint/publishing/soap/FetchPanelsInformationByUrl
```

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

3.1.4.2.1.2   FetchPanelsInformationByUrlSoapOut

The response WSDL message for a **FetchPanelsInformationByUrl** WSDL operation.

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

3.1.4.2.2 Elements

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

Element	Description
FetchPanelsInformationByUrl	The input data for a **FetchPanelsInformationByUrl** WSDL operation.
FetchPanelsInformationByUrlResponse	The result data for a **FetchPanelsInformationByUrl** WSDL operation.

3.1.4.2.2.1 FetchPanelsInformationByUrl

The input data of a **FetchPanelsInformationByUrl** WSDL operation.

```
<s:element name="FetchPanelsInformationByUrl">
  <s:complexType>
    <s:sequence>
      <s:element name="pageLayoutUrl" type="s:string"/>
      <s:element name="cultureName" type="s:string"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

pageLayoutUrl: The URL for a **page layout**. It MUST be a URL as specified in [RFC1738].

cultureName: The **culture name**, as specified in [RFC4646] and [RFC4647], that is used to localize data in the response WSDL message.

3.1.4.2.2.2 FetchPanelsInformationByUrlResponse

The result data of a **FetchPanelsInformationByUrl** WSDL operation.

```
<s:element name="FetchPanelsInformationByUrlResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="FetchPanelsInformationByUrlResult" type="tns:ArrayOfPanelInfo"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

FetchPanelsInformationByUrlResult: An array of toolbox panel information.

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
ArrayOfPanelInfo	Defines an array that contains information about a set of toolbox panels.
PanelInfo	Specifies information about a toolbox panel.

3.1.4.2.3.1 ArrayOfPanelInfo

The **ArrayOfPanelInfo** complex type defines an array that contains information about a set of toolbox panels.

```
<s:complexType name="ArrayOfPanelInfo">
  <s:sequence>
    <s:element name="PanelInfo" type="tns:PanelInfo" minOccurs="2" maxOccurs="2"/>
  </s:sequence>
</s:complexType>
```

PanelInfo: Information about a toolbox panel.

3.1.4.2.3.2 PanelInfo

The **PanelInfo** complex type specifies information about a toolbox panel.

```
<s:complexType name="PanelInfo">
  <s:sequence>
    <s:element name="displayName" type="s:string"/>
    <s:element name="controlListId" type="s:string"/>
    <s:element name="panelTypeIdentifier" type="s:int"/>
  </s:sequence>
</s:complexType>
```

displayName: The display name of the toolbox panel.

controlListId: The identifier of the toolbox panel. The value MUST be a content type identifier.

panelTypeIdentifier: The type of toolbox panel. It MUST be one of the following values:

Value	Meaning
0	The information is about the shared toolbox panel.
2	The information about the toolbox panel was retrieved by using the content type that is associated with the Web page.

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.

3.1.5   Timer Events

None.

3.1.6   Other Local Events

None.

4   Protocol Examples

In this example, a protocol client displays a list of Web controls that can be added to Web pages that are using a page layout. The protocol client is an application that allows users to edit page layouts. To display the list, the protocol client constructs the following WSDL message, which queries the protocol server for the list of available Web controls:

```
<?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>
    <FetchPanelsInformationByUrl
xmlns="http://schemas.microsoft.com/sharepoint/publishing/soap/">

<pageLayoutUrl>http://www.example.com/_catalogs/masterpage/ArticleLeft.aspx</pageLayoutUrl>
      <cultureName>en-us</cultureName>
    </FetchPanelsInformationByUrl>
  </soap:Body>
</soap:Envelope>
```

The protocol server generates information for the two toolbox panels that are associated with the URL specified in the **pageLayoutUrl** element and returns the following **SOAP message**:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <FetchPanelsInformationByUrlResponse
xmlns="http://schemas.microsoft.com/sharepoint/publishing/soap/">
      <FetchPanelsInformationByUrlResult>
        <PanelInfo>
          <displayName>Page Controls</displayName>
          <controlListId>0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007
              948130EC3DB064584E219954237AF39</controlListId>
          <panelTypeIdentifier>0</panelTypeIdentifier>
        </PanelInfo>
        <PanelInfo>
          <displayName>Article Page Controls</displayName>
          <controlListId>0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007
          948130EC3DB064584E219954237AF3900242457EFB8B24247815D688C5
          26CD44D</controlListId>
          <panelTypeIdentifier>2</panelTypeIdentifier>
        </PanelInfo>
      </FetchPanelsInformationByUrlResult>
    </FetchPanelsInformationByUrlResponse>
  </soap:Body>
</soap:Envelope>
```

The protocol client then retrieves the list of Web controls that are associated with the shared toolbox panel by sending the following SOAP message:

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

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <FetchControlLists xmlns="http://schemas.microsoft.com/sharepoint/publishing/soap/">
      <controlListIds>
        <string>0x010100C568DB52D9D0A14D9B2FDCC96666E9F2007
          948130EC3DB064584E219954237AF39</string>
      </controlListIds>
      <cultureName>en-us</cultureName>
    </FetchControlLists>
  </soap:Body>
</soap:Envelope>
```

The protocol server determines that fifteen Web controls are associated with the shared toolbox panel and generates the following SOAP 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>
    <FetchControlListsResponse
xmlns="http://schemas.microsoft.com/sharepoint/publishing/soap/">
      <FetchControlListsResult>
        <string>&lt;?xml version="1.0" encoding="utf-16"?&gt;&lt;ControlsList&gt;&lt;Assembly
Name="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" Namespace="Microsoft.SharePoint.WebControls"
TagPrefix="wss"&gt;&lt;Control Name="Content Type" Title="Content Type" Description="The
Content Type field is a Computed Type field. Insert this control onto your page to include
this field in your page layout." Template="&amp;lt;{0}:ComputedField
FieldName=&amp;quot;ContentType&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:ComputedField&amp;gt;" /&gt;&lt;Control
Name="Name" Title="Name" Description="The Name field is a File Type field. Insert this
control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:FileField FieldName=&amp;quot;FileLeafRef&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:FileField&amp;gt;" /&gt;&lt;Control
Name="Title" Title="Title" Description="The Title field is a Text Type field. Insert this
control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:TextField FieldName=&amp;quot;Title&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:TextField&amp;gt;" /&gt;&lt;Control
Name="Document Modified By" Title="Document Modified By" Description="The Document Modified
By field is a Text Type field. Insert this control onto your page to include this field in
your page layout." Template="&amp;lt;{0}:TextField
FieldName=&amp;quot;Modified_x0020_By&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:TextField&amp;gt;" /&gt;&lt;Control
Name="Document Created By" Title="Document Created By" Description="The Document Created By
field is a Text Type field. Insert this control onto your page to include this field in your
page layout." Template="&amp;lt;{0}:TextField FieldName=&amp;quot;Created_x0020_By&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:TextField&amp;gt;" /&gt;&lt;Control
Name="Description" Title="Description" Description="The Description field is a Note Type
field. Insert this control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:NoteField FieldName=&amp;quot;Comments&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:NoteField&amp;gt;"
/&gt;&lt;/Assembly&gt;&lt;Assembly Name="Microsoft.SharePoint.Publishing, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c"
Namespace="Microsoft.SharePoint.Publishing.WebControls" TagPrefix="cmsc"&gt;&lt;Control
Name="Scheduling Start Date" Title="Scheduling Start Date" Description="The Scheduling Start
Date f
ield is a PublishingScheduleStartDateFieldType Type field. Insert this control onto your page
to include this field in your page layout."
```

```
Template="&amp;lt;{0}:PublishingScheduleFieldControl
FieldName=&amp;quot;PublishingStartDate&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:PublishingScheduleFieldControl&amp;gt;"
/&gt;&lt;Control Name="Scheduling End Date" Title="Scheduling End Date" Description="The
Scheduling End Date field is a PublishingScheduleEndDateFieldType Type field. Insert this
control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:PublishingScheduleFieldControl
FieldName=&amp;quot;PublishingExpirationDate&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:PublishingScheduleFieldControl&amp;gt;"
/&gt;&lt;/Assembly&gt;&lt;Assembly Name="Microsoft.SharePoint, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c"
Namespace="Microsoft.SharePoint.WebControls" TagPrefix="wss"&gt;&lt;Control Name="Contact"
Title="Contact" Description="The Contact field is a User Type field. Insert this control onto
your page to include this field in your page layout." Template="&amp;lt;{0}:UserField
FieldName=&amp;quot;PublishingContact&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:UserField&amp;gt;" /&gt;&lt;Control
Name="Contact E-Mail Address" Title="Contact E-Mail Address" Description="The Contact E-Mail
Address field is a Text Type field. Insert this control onto your page to include this field
in your page layout." Template="&amp;lt;{0}:TextField
FieldName=&amp;quot;PublishingContactEmail&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:TextField&amp;gt;" /&gt;&lt;Control
Name="Contact Name" Title="Contact Name" Description="The Contact Name field is a Text Type
field. Insert this control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:TextField FieldName=&amp;quot;PublishingContactName&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:TextField&amp;gt;" /&gt;&lt;Control
Name="Contact Picture" Title="Contact Picture" Description="The Contact Picture field is a
URL Type field. Insert this control onto your page to include this field in your page
layout." Template="&amp;lt;{0}:UrlField
FieldName=&amp;quot;PublishingContactPicture&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:UrlField&amp;gt;" /&gt;&lt;Control
Name="Page Layout" Title="Page Layout" Description="The Page Layout field is a URL Type
field. Insert this control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:UrlField FieldName=&amp;quot;PublishingPageLayout&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:UrlField&amp;gt;"
/&gt;&lt;/Assembly&gt;&lt;Assembly Name="Microsoft.SharePoint.Publishing, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c"
Namespace="Microsoft.SharePoint.Publishing.WebControls" TagPrefix="cmsc"&gt;&lt;Control
Name="Rollup Image" Title="Rollup Image" Description="The Rollup Image field is a Image Type
field. Insert this control onto your page to include this field in your page layout."
Template="&amp;lt;{0}:RichImageField FieldName=&amp;quot;PublishingRollupImage&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:RichImageField&amp;gt;"
/&gt;&lt;/Assembly&gt;&lt;Assembly Name="Microsoft.Office.Server.UserProfiles,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
Namespace="Microsoft.Office.Server.WebControls.FieldTypes"
TagPrefix="CustomTag_0"&gt;&lt;Control Name="Target Audiences" Title="Target Audiences"
Description="The Target Audiences field is a TargetTo Type field. Insert this control onto
your page to include this field in your page layout."
Template="&amp;lt;{0}:SPFieldTargetToControl FieldName=&amp;quot;Audience&amp;quot;
runat=&amp;quot;server&amp;quot;&amp;gt;&amp;lt;/{0}:SPFieldTargetToControl&amp;gt;"
/&gt;&lt;/Assembly&gt;&lt;/ControlsList&gt;</string>
      </FetchControlListsResult>
    </FetchControlListsResponse>
  </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 and schema are provided in this appendix.

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/publishing/soap/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://schemas.microsoft.com/sharepoint/publishing/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">This web service is
designed for FrontPage client to use</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/publishing/soap/">
      <s:element name="FetchPanelsInformationByUrl">
        <s:complexType>
          <s:sequence>
            <s:element name="pageLayoutUrl" type="s:string"/>
            <s:element name="cultureName" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="FetchPanelsInformationByUrlResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="FetchPanelsInformationByUrlResult" type="tns:ArrayOfPanelInfo"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfPanelInfo">
        <s:sequence>
          <s:element name="PanelInfo" type="tns:PanelInfo" minOccurs="2" maxOccurs="2"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="PanelInfo">
        <s:sequence>
          <s:element name="displayName" type="s:string"/>
          <s:element name="controlListId" type="s:string"/>
          <s:element name="panelTypeIdentifier" type="s:int"/>
        </s:sequence>
      </s:complexType>
      <s:element name="FetchControlLists">
        <s:complexType>
          <s:sequence>
            <s:element name="controlListIds"/>
              <s:sequence>
                <s:element name="string" type="s:string" maxOccurs="unbounded"/>
              </s:sequence>
            <s:element name="cultureName" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="FetchControlListsResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="FetchControlListsResult">
              <s:complexType>
```

```
                  <s:sequence>
                    <s:element name="string" type="s:string" minOccurs="0"
maxOccurs="unbounded"/>
                  </s:sequence>
                </s:complexType>
              </s:element>
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:schema>
    </wsdl:types>
    <wsdl:message name="FetchPanelsInformationByUrlSoapIn">
      <wsdl:part name="parameters" element="tns:FetchPanelsInformationByUrl" />
    </wsdl:message>
    <wsdl:message name="FetchPanelsInformationByUrlSoapOut">
      <wsdl:part name="parameters" element="tns:FetchPanelsInformationByUrlResponse" />
    </wsdl:message>
    <wsdl:message name="FetchControlListsSoapIn">
      <wsdl:part name="parameters" element="tns:FetchControlLists" />
    </wsdl:message>
    <wsdl:message name="FetchControlListsSoapOut">
      <wsdl:part name="parameters" element="tns:FetchControlListsResponse" />
    </wsdl:message>
    <wsdl:portType
name="CMS_x0020_Content_x0020_Area_x0020_Toolbox_x0020_Info_x0020_serviceSoap">
      <wsdl:operation name="FetchPanelsInformationByUrl">
        <wsdl:input message="tns:FetchPanelsInformationByUrlSoapIn" />
        <wsdl:output message="tns:FetchPanelsInformationByUrlSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="FetchControlLists">
        <wsdl:input message="tns:FetchControlListsSoapIn" />
        <wsdl:output message="tns:FetchControlListsSoapOut" />
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding
name="CMS_x0020_Content_x0020_Area_x0020_Toolbox_x0020_Info_x0020_serviceSoap"
type="tns:CMS_x0020_Content_x0020_Area_x0020_Toolbox_x0020_Info_x0020_serviceSoap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="FetchPanelsInformationByUrl">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/publishing/soap/FetchPanelsInformationByU
rl" style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="FetchControlLists">
        <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/publishing/soap/FetchControlLists"
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="CMS_x0020_Content_x0020_Area_x0020_Toolbox_x0020_Info_x0020_serviceSoap12"
type="tns:CMS_x0020_Content_x0020_Area_x0020_Toolbox_x0020_Info_x0020_serviceSoap">
        <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="FetchPanelsInformationByUrl">
            <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/publishing/soap/FetchPanelsInformationByU
rl" style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="FetchControlLists">
            <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/publishing/soap/FetchControlLists"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
</wsdl:definitions>
```

7   Appendix B: Product Behavior

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

- Microsoft Office SharePoint Server 2007

- Microsoft SharePoint Server 2010

- Microsoft SharePoint Server 2013

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

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

8 Change Tracking

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

9   Index

Release: October 30, 2014