[MS-WPPS]:
Web Part Pages 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 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.

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

1 / 224

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
04/25/2008	0.2	Editorial	Revised and edited the technical content
06/27/2008	1.0	Major	Revised and edited the technical content
10/06/2008	1.01	Editorial	Revised and edited the technical content
12/12/2008	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	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	2.05	Editorial	Changed language and formatting in the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.6	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
01/20/2012	2.7	Minor	Clarified the meaning of the technical content.
04/11/2012	2.7	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.7	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	2.7	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Web Part Pages Web Service Protocol, which enables a client to gather the information necessary to author Web pages that use server resources, and to author execution logic that reacts to changes in the server state.

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

1.1 Glossary

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

GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
language code identifier (LCID)

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

absolute URL
action
assembly
base view
byte order mark
comment
content type
displayed version
document
document library
Extensible Application Markup Language (XAML)
feature
front-end web server
list
list item
page
personal view
public view
register directive
server-relative URL
Simple Object Access Protocol (SOAP)
site
site collection
site-collection relative URL
site-relative URL
SOAP action
SOAP body
SOAP envelope
SOAP fault
Uniform Resource Locator (URL)
Uniform Resource Name (URN)
view
Web Part

Web Part connection
Web Part Page
Web Part zone
Web Part zone identifier
Web Part zone index
Web Services Description Language (WSDL)
workflow
XML namespace
XML schema
XML schema definition (XSD)

The following terms are specific to this document:

data source control: An object that can be added to an ASP.NET webpage and encapsulates the necessary logic to connect to a data source, such as a database or XML file, and that can execute queries or other data-access commands. A data source control can in turn provide data to other controls on that page.

declarative workflow: A workflow that is created with XAML (Extensible Application Markup Language) files and does not require precompiled code to run.

tag prefix: In ASP.NET markup, an identifier that appears before the name of a control and specifies the assembly that contains the control. A tag prefix is assigned on a per-page or per-site basis.

workflow configuration file: An implementation-specific file that is a part of a workflow. The workflow configuration file contains information that is necessary to create a workflow template from the specified workflow markup and rules files, and to associate it with a specific list.

workflow markup file: A file that contains markup to specify the functional behavior of a workflow.

workflow rules file: A file that contains markup to specify the rules and conditions that will be evaluated during the execution of the workflow.

XML decode: A process that is used to convert encoded XML content to its original form.

XML encode: A process that is used to convert XML content from one code to another to use that content without affecting the structure or validity of an XML 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

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

1.2.1 Normative References

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

http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-FPSE] Microsoft Corporation, "FrontPage Server Extensions Remote Protocol Specification".

[MS-WSSCAML] Microsoft Corporation, "Collaborative Application Markup Language (CAML) Structure Specification".

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

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Language", RFC 3066, January 2001, http://www.ietf.org/rfc/rfc3066.txt

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

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

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

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

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

[XMLSCHEMA1] Thompson, H.S., 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

[MSDN-AWPRO] Microsoft Corporation, "ASP.NET Web Page Resources Overview", http://msdn.microsoft.com/en-us/library/ms227427.aspx

[MSDN-BCAUSPD] Microsoft Corporation, "Building Custom Activities for Use in SharePoint Designer 2007", July 2007, http://msdn.microsoft.com/en-us/library/bb629922.aspx

[MSDN-GETXML] Microsoft Corporation, "WebPartPagesWebService.GetXmlDataFromDataSource Method (Web Part Pages)", http://msdn.microsoft.com/en-us/library/ms774756.aspx

[MSDN-Register] Microsoft Corporation, ".NET Framework General Reference: @ Register", http://msdn.microsoft.com/en-us/library/c76dd5k1(VS.90).aspx

[MSDN-RIRF] Microsoft Corporation, "Resources in .Resx File Format", http://msdn.microsoft.com/en-us/library/ekyft91f(VS.90).aspx

[MSDN-UWM] Microsoft Corporation, "Using Workflow Markup", http://msdn.microsoft.com/en-us/library/ms735921.aspx

[MSDN-WCCSWP] Microsoft Corporation, "Walkthrough: Creating Connectable SharePoint Web Parts", http://msdn.microsoft.com/en-us/library/ms469765.aspx

[MSDN-WCSO] Microsoft Corporation, "Workflow Configuration Schema Overview", http://msdn.microsoft.com/en-us/library/bb417439.aspx

[MSDN-WPCDF] Microsoft Corporation, "Web Parts Control Description Files", http://msdn.microsoft.com/en-us/library/ms227561.aspx

[MSDN-WPCO] Microsoft Corporation, "Web Parts Connections Overview", http://msdn.microsoft.com/en-us/library/ms178187.aspx

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

[MS-LISTSWS] Microsoft Corporation, "Lists Web Service Protocol Specification".

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

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

1.3   Protocol Overview (Synopsis)

This protocol provides operations related to authoring **page** constructs and **declarative workflows**. These operations fall into the following fairly independent categories:

- Operations to list, create, modify, and delete **Web Parts** in a **Web Part Page**.

- Operations to connect Web Parts. A **Web Part connection** can bind Web Parts only if the Web Parts properly implement provider and consumer Web Part connection interfaces.

- Operations to perform data retrieval in pure XML format or with **data source control**.

- Operations to manipulate Web Parts and controls. This category also includes general operations for authoring Web Part Pages.

- Operations to author, validate, create, and remove a declarative **workflow (2)** on a **site (2)**.

- Operations to retrieve the list of binary files that contain controls permitted on the server, the list of controls in a binary file, control properties, and property types. These operations also enable the client to pass a portion of a page to the server, which includes updates to the page. The server makes these updates based on its information about the controls and the server environment, and returns data so that a client authoring a page can make changes to the client version of the control as if they were being made on the server.

1.4   Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using **HTTP**, as described in [RFC2616], or **HTTPS**, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol:
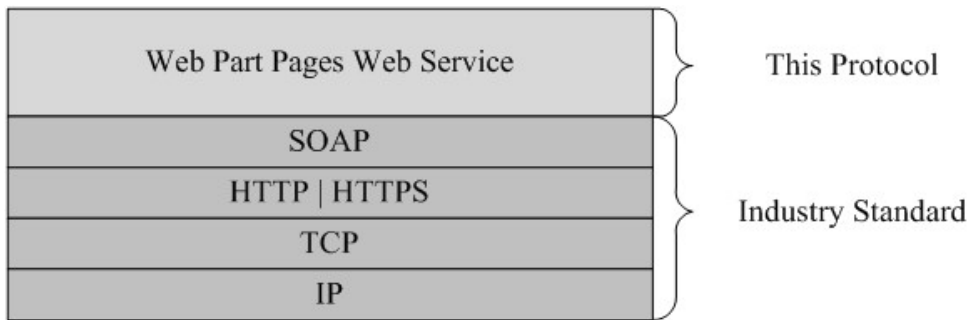
Figure 1: This protocol in relation to other protocols

There is one dependency between this protocol and the [MS-FPSE] protocol. The **GetWebPartPage** and **GetWebPartPageDocument** SOAP operations as provided by this protocol are read operations, and a number of the other SOAP operations provide information to a client necessary to handle the contents of these pages. Therefore, to write a page to the server, the client needs to use an operation that enables placing the contents of a page on the server such as **put documents** in [MS-FPSE] section 3.1.5.3.12.

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/WebPartPages.asmx" to the URL of the site (2), for example: http://www.contoso.com/Repository/_vti_bin/WebPartPages.asmx.

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

1.6   Applicability Statement

This protocol is used by client development tools to author pages and declarative workflows on a **front-end Web server** that supports the protocol. The precise functioning of many of the operations in this protocol depends on the implementation of the server.

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 **Transport** (section 2.1).

- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are described in **Common Message Syntax** (section 2.2) and **Message Processing Events and Sequencing Rules** (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 for securing communication with clients.

Protocol messages MUST be formatted as specified either in [SOAP1.1] section 4, SOAP Envelope or in [SOAP1.2/1] section 5, SOAP Message Construct. Protocol server faults MUST be returned either using HTTP Status Codes as specified in [RFC2616] section 10, Status Code Definitions, or using **SOAP faults** as specified either in [SOAP1.1] section 4.4, SOAP Fault or in [SOAP1.2/1] section 5.4, SOAP Fault.

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 XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://microsoft.com/sharepoint/webpartpages	
s1	http://microsoft.com/wsdl/types/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1], [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1], [SOAP1.2/2]
(none)	http://microsoft.com/sharepoint/webpartpages	
wpv2	http://schemas.microsoft.com/WebPart/v2	
wpv3	http://schemas.microsoft.com/WebPart/v3	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
core	http://schemas.microsoft.com/sharepoint/soap/	[MS-WSSCAML]

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

Element	Description
webParts	Specifies a container for a **wpv3:webPart** element. It is referred to textually as the .webpart format.
XML	Specifies the schema of the data that the server sends in response to a client request to retrieve information about authoring a Web Part connection. It is referred to textually as the Web Part Connection Information Schema.

2.2.3.1 webParts

The **webParts** element specifies a set of Web Parts as follows:

```
<s:schema  xmlns:s="http://www.w3.org/2001/XMLSchema"
   xmlns:wpv3="http://schemas.microsoft.com/WebPart/v3">
   <s:element name="webParts">
      <s:complexType>
         <s:sequence>
            <s:element name="webPart" type="wpv3:webPart" minOccurs="1" maxOccurs="1" />
         </s:sequence>
      </s:complexType>
   </s:element>
</s:schema>
```

webParts: Container for one **wpv3:webPart** element.

webParts.webPart: A **wpv3:webPart** element specifying a single Web Part.

2.2.3.2 XML

The **XML** element specifies a Web Part connection as follows:

```
<s:schema elementFormDefault="qualified" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s1="http://microsoft.com/wsdl/types/">
   <s:import namespace="http://microsoft.com/wsdl/types/" />
   <s:element name="XML">
      <s:complexType>
         <s:sequence>
            <s:element name="ConnDesign">
               <s:complexType>
                  <s:sequence>
                     <s:element name="ConnectionInfo">
                        <s:complexType>
                           <s:sequence>
                              <s:element name="WebPart" minOccurs="0" maxOccurs="unbounded">
                                 <s:complexType>
                                    <s:sequence>
                                       <s:element name="InterfaceGroups" minOccurs="1"
maxOccurs="1">

                                          <s:complexType>
```

```xml
                                            <s:sequence>
                                               <s:element name="InterfaceGroup"
minOccurs="0" maxOccurs="unbounded">
                                                  <s:complexType>
                                                     <s:sequence>
                                                        <s:element name="Interfaces">
                                                           <s:complexType>
                                                              <s:sequence>
                                                                 <s:element
name="Interface">
                                                                    <s:complexType>
                                                                       <s:choice
minOccurs="0" maxOccurs="1">
                                                                          <s:element
name="InitEventArgs" type="InitEventArgsType"/>
                                                                          <s:element
name="Properties" type="PropertiesType" />
                                                                       </s:choice>
                                                                       <s:attribute
name="IsASP" type="s:string" />
                                                                       <s:attribute
name="Match" type="s:string" />
                                                                       <s:attribute
name="InterfaceName" type="s:string" />
                                                                       <s:attribute
name="Type" type="s:string" />
                                                                       <s:attribute
name="MenuLabel" type="s:string" />
                                                                       <s:attribute
name="Description" type="s:string" />
                                                                    </s:complexType>
                                                                 </s:element>
                                                              </s:sequence>
                                                           </s:complexType>
                                                        </s:element>
                                                     </s:sequence>
                                                     <s:attribute name="GroupName"
type="s:string" use="required" />
                                                     <s:attribute name="GroupMenuLabel"
type="s:string" use="required" />
                                                     <s:attribute name="IsASP"
type="s:string" use="required" />
                                                     <s:attribute name="Description"
type="s:string" use="required" />
                                                     <s:attribute name="RunAt"
type="runAtType" use="required" />
                                                     <s:attribute name="MaxConnections"
use="required">
                                                        <s:simpleType>
                                                           <s:restriction base="s:integer">
                                                              <s:enumeration value="-1" />
                                                              <s:enumeration value="1" />
                                                           </s:restriction>
                                                        </s:simpleType>
                                                     </s:attribute>
                                                     <s:attribute
name="AllowCrossPageConnections" type="s:string" use="required" />
                                                     <s:attribute name="ConnectionType"
type="ConnectionTypeType" use="required" />
                                                  </s:complexType>
                                               </s:element>
```

```xml
                                    </s:sequence>
                                </s:complexType>
                            </s:element>
                        </s:sequence>
                        <s:attribute name="ID" type="s:string" />
                        <s:attribute name="WebPartID" type="s1:guid" />
                        <s:attribute name="ConnectionID" type="s1:guid" />
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="Compatibility">
        <s:complexType>
            <s:sequence>
                <s:element name="sWebPart" minOccurs="0" maxOccurs="1">
                    <s:complexType>
                        <s:sequence>
                            <s:element name="sInterfaces">
                                <s:complexType>
                                    <s:sequence>
                                        <s:element name="int" minOccurs="0"
maxOccurs="unbounded">
                                            <s:complexType>
                                                <s:choice minOccurs="0" maxOccurs="1">
                                                    <s:element name="InitEventArgs"
type="InitEventArgsType"/>
                                                    <s:element name="Properties"
type="PropertiesType" />
                                                </s:choice>
                                                <s:attribute name="IsASP"
type="s:string" />
                                                <s:attribute name="id" type="s:string"
/>
                                            </s:complexType>
                                        </s:element>
                                    </s:sequence>
                                </s:complexType>
                            </s:element>
                            <s:element name="sg" minOccurs="0"
maxOccurs="unbounded">
                                <s:complexType>
                                    <s:sequence>
                                        <s:element name="tParts">
                                            <s:complexType>
                                                <s:sequence>
                                                    <s:element name="tWebPart"
minOccurs="0" maxOccurs="unbounded">
                                                        <s:complexType>
                                                            <s:sequence>
                                                                <s:element name="tg"
minOccurs="0" maxOccurs="unbounded">
                                                                    <s:complexType>
                                                                        <s:sequence>
                                                                            <s:element
name="tInterface">
                                                                                <s:complexType>
                                                                                    <s:sequence>
                                                                                        <s:choice
minOccurs="0" maxOccurs="1">
```

```
<s:element name="InitEventArgs" type="InitEventArgsType"/>

<s:element name="Properties" type="PropertiesType" />

</s:choice>

<s:element name="mi" minOccurs="1" maxOccurs="1">

<s:complexType>

<s:sequence />

<s:attribute name="xInfo" type="s:string" />

<s:attribute name="id" type="s:string" />

<s:attribute name="isXFormUINeeded" type="s:string" />

</s:complexType>

</s:element>

<s:element name="xForm" minOccurs="0" maxOccurs="1">

<s:complexType>

<s:sequence />

<s:attribute name="type" type="s:string" />

</s:complexType>

</s:element>

</s:sequence>
                                                                        <s:attribute
name="IsASP" type="s:string" />
                                                                        <s:attribute
name="id" type="s:string" />

</s:complexType>
                                                                    </s:element>
                                                                </s:sequence>
                                                                <s:attribute
name="id" type="s:string" />
                                                                <s:attribute
name="gLabel" type="s:string" />
                                                                <s:attribute
name="des" type="s:string" />
                                                                <s:attribute
name="isMaxedOut" type="s:string" />
                                                                <s:attribute
name="runAt" type="runAtType">
                                                                </s:attribute>
                                                                <s:attribute
name="ConnectionType" type="ConnectionTypeType">
                                                                </s:attribute>
                                                                <s:attribute
name="IsASP" type="s:string" />
                                                                <s:attribute
name="isConn" type="s:string" />
```

```xml
                                                                </s:complexType>
                                                              </s:element>
                                                            </s:sequence>
                                                            <s:attribute name="ID"
type="s:string" use="required" />
                                                            <s:attribute name="WebPartID"
type="s:string" use="optional" />
                                                            <s:attribute
name="ConnectionID" type="s:string" use="optional" />
                                                            <s:attribute name="pinnedTo"
type="runAtType" />
                                                            <s:attribute name="title"
type="s:string" use="required" />
                                                          </s:complexType>
                                                        </s:element>
                                                      </s:sequence>
                                                    </s:complexType>
                                                  </s:element>
                                                </s:sequence>
                                                <s:attribute name="des" type="s:string"
use="required" />
                                                <s:attribute name="isMaxedOut" type="s:string"
 />
                                                <s:attribute name="id" type="s:string"
use="required" />
                                                <s:attribute name="ConnectionType"
type="ConnectionTypeType" use="required" />
                                                <s:attribute name="IsASP" type="s:string" />
                                                <s:attribute name="runAt" type="runAtType"
use="required" />
                                                <s:attribute name="gLabel" type="s:string"
use="required" />
                                              </s:complexType>
                                            </s:element>
                                          </s:sequence>
                                          <s:attribute name="ID" type="s:string" use="required" />
                                          <s:attribute name="pinnedTo" type="runAtType"
use="optional" />
                                          <s:attribute name="WebPartID" type="s:string"
use="optional" />
                                          <s:attribute name="ConnectionID" type="s:string"
use="optional" />
                                          <s:attribute name="title" type="s:string" />
                                      </s:complexType>
                                  </s:element>
                                </s:sequence>
                            </s:complexType>
                        </s:element>
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:sequence>
        <s:attribute name="id" use="optional">
            <s:simpleType>
                <s:restriction base="s:string">
                    <s:enumeration value="MSOConn_Compatibility" />
                </s:restriction>
            </s:simpleType>
        </s:attribute>
    </s:complexType>
```

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

XML: The root node of the Web Part Connection Information Schema.

XML.id: Specifies that this is used for Web Part connection information.

XML.ConnDesign: The container for **ConnectionInfo** and **Compatibility** child elements.

XML.ConnDesign.ConnectionInfo: The container for **WebPart** nodes that can potentially participate in the Web Part connection.

XML.ConnDesign.ConnectionInfo.WebPart: Specifies a single Web Part and all information regarding the Web Part connection interfaces that it supports.

XML.ConnDesign.ConnectionInfo.WebPart.ID: The control name that identifies this element in the page in which it is located.

XML.ConnDesign.ConnectionInfo.WebPart.WebPartID: A **GUID** that uniquely identifies this element in the page in which it is located.

XML.ConnDesign.ConnectionInfo.WebPart.ConnectionID: A GUID that identifies the Web Part that this element references. It is unique in the scope of both same-page Web Part connections and cross-page Web Part connections.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups: A container for all Web Part connection interface groups supported by the parent **WebPart** element.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup: Specifies a single interface group that the ancestor **WebPart** element supports. An interface group contains either zero or one interface. An interface refers to one manner in which the ancestor **WebPart** element can connect to another Web Part.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.GroupName: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.GroupMenuLabel: A localized string that is a user-recognizable short name for the interface group.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.IsASP: Specifies whether this element can have a descendent element of type **InitEventArgs** or type **PropertiesType**.[1]

Value	Description
True	This **InterfaceGroup** element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	This **InterfaceGroup** element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Description: A localized description of the interface group.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.runAt: Specifies whether the Web Part connection can run on the server that is rendering the page, on the client receiving the rendered page, or on either.

23 / 224

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.MaxConnectio ns: Specifies the maximum number of Web Part connections supported by this interface group. A value of -1 indicates that there is no limit.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.AllowCrossPag eConnections: Specifies whether this interface group supports cross-page Web Part connections.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.ConnectionTyp e: Specifies whether this interface group is a provider or a consumer of data in a Web Part connection.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces: The container for the **Interface** members of the parent single **InterfaceGroup**.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface: Specifies a single Web Part connection interface that is a member of the ancestor Web Part connection interface group that is implemented by the ancestor Web Part. If **IsASP** is set to "True", it can have a **Properties** child element. If **IsASP** is set to "False", it can have an **InitEventArgs** child element.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.IsASP: Specifies whether this element can have a descendent element of type **InitEventArgs** or type **PropertiesType**.<2>

Value	Description
True	This **Interface** element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	This **Interface** element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.Match: The server MUST return this set to True.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.InterfaceName: Identifies an instance of using a Web Part connection interface on a Web Part. If the moniker is undefined in the object type of the Web Part, the server MUST retrieve this value from the definition of the type of the Web Part connection.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.Type: The name of the object type in the object-oriented system that names the interface that the Web Part connection implements.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.MenuLabel: A localized string that is a user-recognizable short name for the Web Part connection interface.

XML.ConnDesign.ConnectionInfo.WebPart.InterfaceGroups.InterfaceGroup.Interfaces.Int erface.Description: A localized description of the interface.

XML.ConnDesign.Compatibility: The container for the **sWebPart** element.

XML.ConnDesign.Compatibility.sWebPart: Specifies the Web Part connection details for a single Web Part that can form a Web Part connection with one of the descendant **tWebPart** elements.

XML.ConnDesign.Compatibility.sWebPart.ID: A control name that uniquely identifies the **sWebPart** in the page in which it is located.

XML.ConnDesign.Compatibility.sWebPart.WebPartID: A GUID that uniquely identifies this element in the page in which it is located.

XML.ConnDesign.Compatibility.sWebPart.ConnectionID: A GUID that identifies the Web Part that this element references. It is unique in the scope of both same-page Web Part connections and cross‑page Web Part connections.

XML.ConnDesign.Compatibility.sWebPart.title: A string that is the friendly name for the Web Part.

XML.ConnDesign.Compatibility.sWebPart.pinnedTo: Specifies whether this Web Part processes its connections on the client or on the server, given the current state of the authoring of the page.

XML.ConnDesign.Compatibility.sWebPart.sInterfaces: A container for the **int** elements that specify the interfaces available on the Web Part referred to by the ancestor **sWebPart**.

XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int: Specifies a single interface that is available for a Web Part connection on the Web Part referred to by the ancestor **sWebPart**.

XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int.IsASP: Specifies whether this element can have a descendent element of type **InitEventArgs** or type **PropertiesType**.<3> If absent, the client MUST interpret this as being set to FALSE.

Value	Description
True	This element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	This element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int.id: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg: Specifies a Web Part connection interface group on the Web Part referenced by the parent **sWebPart** element. This element contains the possible combinations of Web Part connections between the referenced Web Part and other Web Parts in the target scope of the Web Part connection interface group that is specified by this element.

XML.ConnDesign.Compatibility.sWebPart.sg.des: A localized description of the interface group.

XML.ConnDesign.Compatibility.sWebPart.sg.isMaxedOut: Indicates whether the connection has exceeded its maximum number of allowed connections.

XML.ConnDesign.Compatibility.sWebPart.sg.id: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg.ConnectionType: Specifies whether this interface group is a provider or a consumer of data in a Web Part connection.

XML.ConnDesign.Compatibility.sWebPart.sg.IsASP: Specifies whether the **XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int** element with the same **id** as this element and a common **sWebPart** ancestor with this element is permitted to have a descendent element of type **InitEventArgs** or type **PropertiesType**.<4>

Value	Description
True	The related **XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int** element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	The related **XML.ConnDesign.Compatibility.sWebPart.sInterfaces.int** element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.Compatibility.sWebPart.sg.runAt: Specifies whether the connection can run on the server that is rendering the page, the client receiving the rendered page, or on either.

XML.ConnDesign.Compatibility.sWebPart.sg.gLabel: A localized string that is a user-recognizable short name for the interface group.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts: A container for the **tWebPart** elements that are Web Parts that implement a Web Part connection to which the Web Part connection interface group of the **sg** element can connect.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart: Specifies a single Web Part that supports one interface connection group with which Web Part connection interface group of the ancestor **sg** can form a Web Part connection.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.ID: A control name that uniquely identifies the **tWebPart** in the page in which it is located.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.WebPartID: A GUID that uniquely identifies this element in the page in which it is located.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.ConnectionID: A GUID that identifies the Web Part that this element references. It is unique in the scope of both same-page Web Part connections and cross-page Web Part connections.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.pinnedTo: Specifies whether this Web Part processes its connections on the client or on the server, given the current state of the authoring of the page.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.title: A string that is the friendly name for the Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg: Specifies a single connection interface group that is supported by the parent **tWebPart**. An interface refers to one manner in which the ancestor **tWebPart** can participate in Web Part connections.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.isConn: Set to True when the Web Part has at least one Web Part connection. Otherwise, set to False.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.des: A localized description of the interface group.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.isMaxedOut: Specifies whether the connection has exceeded its maximum number of allowed Web Part connections.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.id: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.ConnectionType: Specifies whether this interface group is a provider or a consumer of data in a Web Part connection.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.IsASP: Specifies whether this element can have a descendent element of type **InitEventArgs** or type **PropertiesType**.<5>

Value	Description
True	This element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	This element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.runAt: Specifies whether the connection can run on the server that is rendering the page, on the client receiving the page, or on either.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.gLabel: A localized string that is a user-recognizable short name for the interface group.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.tInterface: Specifies a single connection interface that is part of the parent **tg** connection interface group. If **IsASP** is set to True, it can have a **Properties** child element. If **IsASP** is set to False, it can have an **InitEventArgs** child element.

XML.ConnDesign.Compatibility.sWebPart. sg.tParts.tWebPart.tg.tInterface.IsASP: Specifies whether this element can have a descendent element of type **InitEventArgs** or type **PropertiesType**.<6>

Value	Description
True	This element MUST NOT have a descendant of type **InitEventArgs**.
False (omitted)	This element MUST NOT have a descendant of type **PropertiesType**.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.tInterface.id: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.mi: Specifies a single interface that matches the Web Part connection interface of the ancestor **sg** element. This element contains information necessary to author a transformation between the provider and the consumer, if required.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.mi.xInfo: Specifies implementation-specific<7> details about authoring a transformation between the provider and the consumer.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.mi.id: Identifies an instance of using a Web Part connection interface on a Web Part.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.mi.isXFormUINeeded: Specifies whether special UI is needed to specify the transformation of data from one provider interface to a nominally incompatible consumer interface.

XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.xForm.type: Specifies implementation-specific<8> name for an object that can transform data for a provider interface into data required by a different consumer interface.

2.2.4 Complex Types

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

Complex type	Description
SOAPFaultDetails	Details of a SOAP fault.
wpv2:WebPart	Specifies the schema of a single Web Part object that is either sent to the server or returned from the server. It is referred to textually as the Web Part Definition (DWP) format.
wpv3:webPart	Specifies the schema of a single Web Part object that is either sent to the server or returned from the server.
InitEventArgsType	The implementation-specific XML that specifies the parameters required by a Web Part connection to connect two Web Parts.
PropertiesType	The **Property** elements that specify the set of properties to be passed as part of a Web Part connection.

2.2.4.1 SOAPFaultDetails

The **SOAPFaultDetails** complex type specifies the details of a SOAP fault. This complex type is defined as follows:

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap">
   <s:complexType name="SOAPFaultDetails">
      <s:sequence>
         <s:element name="errorstring" type="s:string" />
         <s:element name="errorcode" type="s:string" minOccurs="0" />
      </s:sequence>
   </s:complexType>
</s:schema>
```

errorString: A human-readable text string explaining the application-level fault.

errorCode: The hexadecimal representation of a 4-byte result code.

2.2.4.2 wpv2:WebPart

The **WebPart** element specifies a Web Part.

A Web Part describes a type in an object-oriented system that is derived from a type that represents a control. As in all types in object-oriented systems, it can contain sub-objects of types in the object-oriented system. The DWP format encompasses the serialization of a Web Part as XML. The contained sub-objects of the Web Part are also serialized. This element is defined as follows:

```
<s:schema targetNamespace="http://schemas.microsoft.com/WebPart/v2"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:wpv2="http://schemas.microsoft.com/WebPart/v2" elementFormDefault="qualified">
   <s:element name="WebPart">
      <s:complexType>
```

```
                <s:sequence minOccurs="2" maxOccurs="unbounded">
                    <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
                </s:sequence>
                <s:attribute name="ID" type="s:string" use="optional" />
                <s:anyAttribute namespace="##any"/>
            </s:complexType>
        </s:element>
    </s:schema>
```

WebPart: The presence of this element specifies the existence of the Web Part.

WebPart.ID: The identifier of the Web Part. It MUST be unique in a single page.

WebPart.*: Each of these elements represents one sub-object of the Web Part. If the server can represent the type of the sub-object in a single text value, this element contains exactly one child node, which MUST be a text node. If the server is unable to represent the type as a single string, this element has sub-elements that represent sub-objects of the sub-object. The nesting of these nodes is permitted. Two elements, **WebPart.Assembly** and **WebPart.TypeName,** have particular meaning, as specified later.

WebPart.Assembly: The text node child of this element specifies the binary file on the front-end Web server that contains the type definition in the object-oriented system for the Web Part.

WebPart.TypeName: The text node child of this element specifies the name of the type in the object-oriented system.

WebPart.*: Each of these attributes is an additional attribute on the **WebPart** tag. The meaning of these attributes is implementation-specific.<9>

2.2.4.3   wpv3:webPart

The **webPart** complex type specifies a Web Part as follows:

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://schemas.microsoft.com/WebPart/v3"
    xmlns:wpv3="http://schemas.microsoft.com/WebPart/v3">
  <s:complexType name="propertyContainerType">
    <s:sequence>
      <s:choice>
        <s:element name="property" type="wpv3:propertyType" minOccurs="0"
maxOccurs="unbounded" />
        <s:element name="ipersonalizable" minOccurs="0" maxOccurs="1">
          <s:complexType>
            <s:sequence>
              <s:element name="property" type="wpv3:propertyType" minOccurs="0"
maxOccurs="unbounded" />
            </s:sequence>
          </s:complexType>
        </s:element>
      </s:choice>
    </s:sequence>
  </s:complexType>
  <s:complexType name="webPart">
    <s:sequence>
      <s:element name="metaData" minOccurs="1" maxOccurs="1">
        <s:complexType>
          <s:sequence>
```

```xml
                        <s:element name="type" minOccurs="1" maxOccurs="1">
                           <s:complexType>
                              <s:sequence>
                              </s:sequence>
                              <s:attribute name="name" type="s:string">
                              </s:attribute>
                              <s:attribute name="src" type="s:string">
                              </s:attribute>
                           </s:complexType>
                        </s:element>
                        <s:element name="importErrorMessage" type="s:string" minOccurs="1"
  maxOccurs="1">
                        </s:element>
                     </s:sequence>
                  </s:complexType>
               </s:element>
               <s:element name="data" minOccurs="1" maxOccurs="1">
                  <s:complexType>
                     <s:sequence>
                        <s:element name="properties" type="wpv3:propertyContainerType"
  minOccurs="1" maxOccurs="1"/>
                        <s:element name="genericWebPartProperties"
  type="wpv3:propertyContainerType" minOccurs="0" maxOccurs="1"/>
                     </s:sequence>
                  </s:complexType>
               </s:element>
            </s:sequence>
         </s:complexType>
         <s:complexType name="propertyType">
            <s:attribute name="name" type="s:string" use="required" />
            <s:attribute name="type" type="s:string" use="required" />
            <s:attribute name="null" type="s:boolean" />
         </s:complexType>
      </s:schema>
```

webPart: Specifies one instance of a Web Part with the descendant elements specifying the type of Web Part and its instance properties.

webPart.metaData: Specifies information related to the type of the Web Part.

webPart.metaData.type: The attributes of this element determine the type of the Web Part. If both are set, **name** is used. If only one is set, that is used. If neither is set, this element is not valid.

webPart.metaData.type.name: Specifies an implementation-specific<10> string to determine a binary file name and type name for the type of the Web Part.

webPart.metaData.type.src: Specifies an implementation-specific<11> string to determine the location of a text file that contains the implementation of the Web Part.

webPart.metaData.importErrorMessage: Specifies the error message that the server MUST use if it is unable to successfully generate the instance of the Web Part defined by this element.

webPart.data: This element and its descendants specify the instance-specific properties of the Web Part.

webPart.data.properties: The descendant elements determine the instance properties of the Web Part.

webPart.data.genericWebPartProperties: The descendant elements determine the instance properties of the Web Part.

propertyType: Specifies one property as a name-value pair for the instance of the Web Part. The child text node defines the value of the property.

propertyType.name: Specifies the name of the property.

propertyType.type: Specifies an implementation-specific<12> string for the type of the property.

propertyType.null: If the type can have a null value, then if this attribute is set to TRUE, it specifies that the value is null.

propertyContainerType: Specifies a set of properties from which to instantiate a Web Part.

propertyContainerType.property: Specifies one property of the Web Part.

propertyContainerType.ipersonalizable: Specifies a set of properties of the Web Part that can be customized by the user who views the Web Part.

propertyContainerType.ipersonalizable.property: Specifies one property that can be configured for each user.

2.2.4.4   InitEventArgs

The **InitEventArgs** element specifies the parameters required by a Web Part connection as follows:

```
<s:complexType name="InitEventArgsType" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:sequence>
      <s:any namespace="##local" processContents="skip" minOccurs="0" maxOccurs="unbounded"
/>
   </s:sequence>
   <s:attribute name="Type" use="optional">
      <s:simpleType>
         <s:restriction base="s:string">
            <s:enumeration value="Client" />
            <s:enumeration value="Server" />
         </s:restriction>
      </s:simpleType>
   </s:attribute>
</s:complexType>
```

InitEventArgsType: Specifies the information necessary to fire a Web Part connection.

InitEventArgsType.*: The implementation-specific XML<13> that specifies the parameters required by a Web Part connection to connect two Web Parts together.

InitEventArgsType.Type: Specifies whether the Web Part connection shares data between the Web Parts on the server processing the page or on the client rendering the page. If not present, the value "Client" is implied. If there are no sub-elements defining the Web Part connection details of values required to be passed, the value of this attribute MUST be set to "Client". If there are sub-elements defining the values that are required to be passed, this value is the same as the **runAtType** type of the ancestor element.

Value	Description
Client	The connection runs on the client, or there are no parameters to pass.
Server	The connection runs on the server.

2.2.4.5 PropertiesType

The **PropertiesType** element specifies the properties of a Web Part connection is defined as follows:

```
<s:complexType name="PropertiesType" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:sequence>
      <s:element name="Property" minOccurs="0" maxOccurs="unbounded">
         <s:complexType>
            <s:sequence>
            </s:sequence>
            <s:attribute name="Name" type="s:string" use="required" />
            <s:attribute name="Description" type="s:string" use="optional" />
            <s:attribute name="DisplayName" type="s:string" use="optional" />
         </s:complexType>
      </s:element>
   </s:sequence>
</s:complexType>
```

PropertiesType: The child **Property** elements specify the set of properties that MUST be passed as part of a connection.

PropertiesType.Property: Specifies a key for which a value is passed from the provider to the consumer as part of the processing of a Web Part connection.

PropertiesType.Property.Name: The internal name of the property used at processing time.

PropertiesType.Property.DisplayName: The localized string naming the property.

PropertiesType.Property.Description: The localized string describing the property.

2.2.5 Simple Types

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

Simple type	Description
SPWebServiceBehavior	Specifies the server behavior for a version of the protocol.
Storage	Specifies the **view** of the page to use for this operation.
s1:guid	Specifies a GUID.

2.2.5.1 SPWebServiceBehavior

The **SPWebServiceBehavior** simple type specifies the server behavior for a version of the protocol as follows:

```
<s:simpleType name="SPWebServiceBehavior" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:restriction base="s:string">
      <s:enumeration value="Version3"/>
   </s:restriction>
</s:simpleType>
```

The following table specifies possible values for this element.

Value	Description
Version3	Specifies the server behavior described in this document.

2.2.5.2 Storage

The **Storage** simple type specifies the level of visibility in a Web Part Page of a Web Part in a **Web Part zone**. This simple type is defined as follows:

```
<s:simpleType name="Storage" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:restriction base="s:string">
      <s:enumeration value="None"/>
      <s:enumeration value="Personal"/>
      <s:enumeration value="Shared"/>
   </s:restriction>
</s:simpleType>
```

The following table specifies possible values for this simple type.

Value	Description
None	The server MAY<14> choose to have this operation affect either the current user's **personal view** or the **public view** of the Web Part Page.
Personal	This operation affects only the current user's personal view of the Web Part Page.
Shared	This operation affects the public view of the Web Part Page.

2.2.5.3 s1:guid

The **s1:guid** simpleType specifies a GUID. This simple type is defined as follows:

```
<s:simpleType name="guid" targetNamespace="http://microsoft.com/wsdl/types/" >
   <s:restriction base="s:string">
      <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}" />
   </s:restriction>
</s:simpleType>
```

Specifies a GUID.

2.2.6 Attributes

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

Attribute	Description
ConnectionTypeType	Specifies whether this element is the consumer or the provider in the provider-consumer relationship.
runAtType	Specifies whether a Web Part connection can be processed by the client rendering a page or on the server processing the page.

2.2.6.1 ConnectionTypeType

The **ConnectionTypeType** simple type specifies whether this element is the consumer or the provider in the provider-consumer relationship. This simple type is defined as follows:

```
<s:simpleType name="ConnectionTypeType" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:restriction base="s:string">
      <s:enumeration value="Provider" />
      <s:enumeration value="Consumer" />
   </s:restriction>
</s:simpleType>
```

Value	Description
Provider	This element plays the role of provider in the provider-consumer connection relationship.
Consumer	This element plays the role of the consumer in the provider-consumer connection relationship.

2.2.6.2 runAtType

The **runAtType** simple type specifies whether a Web Part connection can be processed by the client rendering a page or on the server processing the page. This simple type is defined as follows:

```
<s:simpleType name="runAtType" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:restriction base="s:string">
      <s:enumeration value="None" />
      <s:enumeration value="Client" />
      <s:enumeration value="Server" />
      <s:enumeration value="ServerAndClient" />
   </s:restriction>
</s:simpleType>
```

Value	Description
one	The connection does not run on either the client or the server.
Client	The connection runs on the client.
Server	The connection runs on the server.

Value	Description
ServerAndClient	The connection runs on either the client or the server.

2.2.7 Groups

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

2.2.8 Attribute Groups

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

2.2.9 Common Data Structures

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

3   Protocol Details

The client side of this protocol is 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 returned by the protocol server as specified in [RFC2616], section 10, Status Code Definitions.

This protocol enables protocol servers to notify protocol clients of application-level faults, using SOAP faults. This protocol enables protocol servers to provide additional details for SOAP faults by including either a **detail** element as specified in [SOAP1.1], section 4.4, SOAP Fault or a **Detail** element as specified in [SOAP1.2/1], section 5.4.5, SOAP Detail Element, which conforms to the **XML schema definition (XSD)** of the **SOAPFaultDetails** complex type specified in **SOAPFaultDetails** (section 2.2.4.1). Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol enables protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults, using either HTTP Status Codes or SOAP faults, as specified previously in this section.

This protocol enables 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 not significant for interoperability.

An implementation does not conform to this specification if it fails to satisfy one or more of the MUST requirements that are defined herein. A SOAP node cannot use elements and attributes of the declared XML namespace, which is identified on the title page of this specification, within **SOAP envelopes**, unless it conforms to all MUST requirements of the specification. If a SOAP node does not conform to all of the MUST requirements that are defined in this specification, the behavior of the receiver of the SOAP node is undefined and causes unpredictable results.

3.1   WebPartPagesWebServiceSoap 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 specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

Control: The server can have a variety of compiled binary files that implement the behavior of a UI element or other portion of a page. This protocol contains methods that query the set of available Control classes and metadata about the public interfaces of the Control classes. The server also performs operations on that class when processing pages in response to client requests, and therefore a method of storage/registration that is efficient for the particular server implementation will make both the operations of this protocol as well as processing pages efficient in regards to Control handling.

Page: Each page has a specific URL that a client can use to refer to it. The pages optionally contain dynamic markup. A Web Part Page is a type of page. A Web Part Page can contain Web Part zones, which are addressable by operations in this protocol, and can be stored separately in physical storage from the actual page contents to facilitate efficient queries and updates.

Workflow: This protocol operates on declarative workflows that are specified by a set of files. This collection of files consisting of at least a **workflow markup file** and a **workflow configuration file**. A **workflow rules file**, compiled versions of the other workflow files, and forms for initiation of the workflow (2) and data collection can also be included in the collection. These files can be either literal files in the server file system or entries in a database that represent files. The operations in this protocol require that the markup, rules, and configuration files that comprise a workflow (2) be accessible by a URL, but it is not necessary (or recommended) that the files be accessible by unauthenticated Web requests. Data about the associations of the declarative workflow with lists can be stored with the list or in a separate data store. Based on the operations in this protocol, having an internal representation of associations of workflows (2) that allows efficient querying by an identifier and by the URL and version of the workflow configuration file will allow for more efficient functionality within this protocol.

3.1.2 Timers

None.

3.1.3 Initialization

Each operation of the protocol is essentially independent of the functioning of other operations, and no special initialization occurs by calling one operation that affects the functioning of subsequent operations.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the list of operations defined in this specification.

Operation	Description
AddWebPart	Adds a Web Part to a specified view of a page.
AddWebPartToZone	Adds a Web Part to a specified Web Part zone in a specified view of a page.
AssociateWorkflowMarkup	Associates a workflow (2) already on the server with a **list (1)**.
ConvertWebPartFormat	Converts a Web Part from the persistence format to page markup, or vice-versa.
DeleteWebPart	Removes a Web Part from a view of a page.
ExecuteProxyUpdates	Transmits a modification to a page containing controls to the server so that the implementation of the page running in the server environment can evaluate the change.
FetchLegalWorkflowActions	Returns localized UI strings and specifications for the **actions (2)** and conditions available for use in a declarative workflow on the server.
GetAssemblyMetaData	Retrieves a description of objects in an object-oriented system that derive from one or more base object types contained in a binary file on the server.
GetBindingResourceData	Retrieves an implementation-specific list (1) of localized server resources for use in page

Operation	Description
	editing.
GetCustomControlList	Retrieves an implementation-specific list (1) of the controls that are available for use in pages on the server.
GetDataFromDataSourceControl	Retrieves data similar or identical to the data that a data source control would retrieve during processing of a page.
GetExpandedListViewXml	Retrieves data similar to the data on which a view of a list (1) operates when processing a page.
GetFormCapabilityFromDataSourceControl	Retrieves the list of data manipulation operations supported by a data source control.
GetSafeAssemblyInfo	Retrieves the full names of a set of **assemblies** on the server.
GetWebPart	Retrieves the markup of a Web Part from a Web Part zone in a view of a page.
GetWebPart2	Retrieves the markup of a Web Part from a Web Part zone in a view of a page. Accepts an additional parameter to request behavior for a specified protocol version.
GetWebPartCrossPageCompatibility	Returns the available Web Part connections between two Web Parts on different pages.
GetWebPartPage	Retrieves the implementation-specific markup for a page, as well as the metadata of the page. Accepts an additional parameter to request behavior for a specified protocol version.
GetWebPartPageConnectionInfo	Returns the available Web Part connections between two Web Parts on the same page.
GetWebPartPageDocument	Retrieves the implementation-specific markup for a page, as well as the metadata of the page.
GetWebPartProperties	Retrieves the markup for all Web Parts in a view of a page.
GetWebPartProperties2	Retrieves the markup for all Web Parts in a view of a page. Accepts an additional parameter to request behavior for a specified protocol version.
GetXmlDataFromDataSource	Retrieves data from a data store in which the query is in pure XML format.
RemoveWorkflowAssociation	Removes an association of a workflow (2) from a list (1).
RenderWebPartForEdit	Retrieves implementation-specific markup for a single Web Part that can be used to design

Operation	Description
	pages.
SaveWebPart	Updates an existing Web Part in a page.
SaveWebPart2	Updates an existing Web Part in a page. In addition to the functionality of **SaveWebPart**, it enables the client to change the underlying object type of the Web Part.
ValidateWorkflowMarkupAndCreateSupportObjects	Validates a group of workflow (2) files, potentially in addition to further implementation-specific behavior.

3.1.4.1 AddWebPart

The **AddWebPart** operation adds a Web Part to a specified view of a page. It is a simplified version of the **AddWebPartToZone** operation. To specify which Web Part zone to which to add the Web Part, use **AddWebPartToZone** instead. This operation is defined as follows:

```
<wsdl:operation name="AddWebPart" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="AddWebPartSoapIn" />
    <wsdl:output message="AddWebPartSoapOut" />
</wsdl:operation>
```

The client sends an **AddWebPartSoapIn** request message, and the server responds with an **AddWebPartSoapOut** response message.

3.1.4.1.1 Messages

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

Message	Description
AddWebPartSoapIn	A request to initiate a **AddWebPart** operation on the protocol server.
AddWebPartSoapOut	A response from the protocol server at completion of the **AddWebPart** operation.

3.1.4.1.1.1 AddWebPartSoapIn

The **AddWebPartSoapIn** message is the request message for **AddWebPart**.

The **SOAP action** value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/AddWebPart
```

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

3.1.4.1.1.2 AddWebPartSoapOut

The **AddWebPartSoapOut** message is the response message for **AddWebPart**.

The SOAP Body contains an **AddWebPartResponse** element.

3.1.4.1.2 Elements

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

Element	Description
AddWebPart	A request to initiate a **AddWebPartSoapIn** operation on the protocol server.
AddWebPartResponse	A response from the protocol server at completion of the **AddWebPartSoapOut** operation.

3.1.4.1.2.1 AddWebPart

The **AddWebPart** element is defined as follows:

```
<s:element name="AddWebPart" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

The definition of the parameters of this element is identical to that for the **AddWebPartToZone** element. When processing this element, the server MUST respond as if the same parameters were passed to **AddWebPartToZone** and all other parameters of **AddWebPartToZone** were omitted.

3.1.4.1.2.2 AddWebPartResponse

The **AddWebPartResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="AddWebPartResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="AddWebPartResult" type="s1:guid" minOccurs="1" maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

AddWebPartResult: A GUID that uniquely identifies the Web Part usable by subsequent operations to update, retrieve, or delete the Web Part.

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 AddWebPartToZone

The **AddWebPartToZone** operation<15> adds a Web Part to a specified Web Part zone in a specified view of a page. This operation is defined as follows:

```
<wsdl:operation name="AddWebPartToZone" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="AddWebPartToZoneSoapIn" />
   <wsdl:output message="AddWebPartToZoneSoapOut" />
</wsdl:operation>
```

The client sends an **AddWebPartToZoneSoapIn** request message, and the server responds with an **AddWebPartToZoneSoapOut** response message.

3.1.4.2.1 Messages

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

Message	Description
AddWebPartToZoneSoapIn	A request to initiate a **AddWebPartToZone** operation on the protocol server.
AddWebPartToZoneSoapOut	A response from the protocol server at completion of the **AddWebPartToZone** operation.

3.1.4.2.1.1 AddWebPartToZoneSoapIn

The **AddWebPartToZoneSoapIn** message is the request message for **AddWebPartToZone**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/AddWebPartToZone
```

The SOAP Body contains an **AddWebPartToZone** element.

3.1.4.2.1.2 AddWebPartToZoneSoapOut

The **AddWebPartToZoneSoapOut** message is the response message for **AddWebPartToZone**.

The SOAP Body contains an **AddWebPartToZoneResponse** element.

3.1.4.2.2 Elements

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

Element	Description
AddWebPartToZone	A request to initiate a **AddWebPartToZoneSoapIn** operation on the protocol server.
AddWebPartToZoneResponse	A response from the protocol server at completion of the **AddWebPartToZoneSoapOut** operation.

3.1.4.2.2.1 AddWebPartToZone

The **AddWebPartToZone** element is defined as follows:

```
<s:element name="AddWebPartToZone" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="1" maxOccurs="1"/>
         <s:element name="zoneId" type="s:string" minOccurs="0" maxOccurs="1"/>
         <s:element name="zoneIndex" type="s:int" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

pageUrl: The location of the page. MUST be formatted as either an **absolute URL**, a **server-relative URL**, or a **site-relative URL**. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current **site collection**.

webPartXml: An XML-encoded string that, after decoding, MUST be in one of the formats supported by the server.<16> This is the contents of the Web Part being added.

storage: Specifies whether the operation adds Web Parts to the public view or the current user's personal view. MUST be set to either "Shared" or "Personal".

zoneId: The **Web Part zone identifier** to which to add the Web Part.<17> If this element is omitted or refers to a Web Part zone that does not exist in the page, the server can add the Web part to any Web Part zone in the page, but the server MUST add the Web Part to a Web Part zone in the page and SHOULD ignore the **zoneIndex** attribute. If there are no Web Part zones in the page, the server SHOULD<18> store the Web Part, but not add it to the page.

zoneIndex: A non-negative **Web Part zone index** that indicates the position at which the Web Part is to be inserted in the Web Part zone. If this value is omitted, the server MUST treat this as if it were set to zero. The new Web Part MUST be inserted directly before the existing Web Part with the lowest Web Part zone index that is greater than or equal to **zoneIndex**. If there is no such existing Web Part (such as if **zoneIndex** is greater than the index of all Web Parts in the zone or there are

no Web Parts in the Web Part zone), the new Web Part MUST<19> be inserted at the end of the Web Part zone.

3.1.4.2.2.2 AddWebPartToZoneResponse

The **AddWebPartToZoneResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="AddWebPartToZoneResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="AddWebPartToZoneResult" type="s1:guid" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

AddWebPartToZoneResult: A GUID that uniquely identifies the Web Part and can be used to update, retrieve, or delete the Web Part.

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 AssociateWorkflowMarkup

The **AssociateWorkflowMarkup** operation<20> causes a workflow configuration file<21> already present on the server to be processed and associated with a list (1).

Before using this operation, the caller MUST first upload the workflow markup file, workflow rules file, and workflow configuration file for the workflow (2) to associate, and then MUST call **ValidateWorkflowMarkupAndCreateSupportObjects** to validate that workflow (2) and prepare it for running.

To upload documents, use the FrontPage Server Extensions (FPSE) Remote Protocol **put document** method, as specified in [MS-FPSE] section 3.1.5.3.11.

This operation is defined as follows:

```
<wsdl:operation name="AssociateWorkflowMarkup" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="AssociateWorkflowMarkupSoapIn" />
   <wsdl:output message="AssociateWorkflowMarkupSoapOut" />
</wsdl:operation>
```

The client sends an **AssociateWorkflowMarkupSoapIn** request message, and the server responds with an **AssociateWorkflowMarkupSoapOut** response message.

3.1.4.3.1 Messages

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

Message	Description
AssociateWorkflowMarkupSoapIn	A request to initiate a **AssociateWorkflowMarkup** operation on the protocol server.
AssociateWorkflowMarkupSoapOut	A response from the protocol server at completion of the **AssociateWorkflowMarkup** operation.

3.1.4.3.1.1 AssociateWorkflowMarkupSoapIn

The **AssociateWorkflowMarkupSoapIn** message is the request message for **AssociateWorkflowMarkup**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/AssociateWorkflowMarkup
```

The SOAP Body contains an **AssociateWorkflowMarkup** element.

3.1.4.3.1.2 AssociateWorkflowMarkupSoapOut

The **AssociateWorkflowMarkupSoapOut** message is the response message for **AssociateWorkflowMarkup**.

The SOAP Body contains an **AssociateWorkflowMarkupResponse** element.

3.1.4.3.2 Elements

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

Element	Description
AssociateWorkflowMarkup	A request to initiate a **AssociateWorkflowMarkupSoapIn** operation on the protocol server.
AssociateWorkflowMarkupResponse	A response from the protocol server at completion of the **AssociateWorkflowMarkupSoapOut** operation.

3.1.4.3.2.1 AssociateWorkflowMarkup

The **AssociateWorkflowMarkup** element is defined as follows:

```
<s:element name="AssociateWorkflowMarkup" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="configUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="configVersion" minOccurs="0" maxOccurs="1">
            <s:simpleType>
               <s:restriction base="s:string">
                  <s:pattern value="[v|V]\d+\.\d+" />
               </s:restriction>
            </s:simpleType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

configUrl: An unencoded site-relative URL that indicates the location of the workflow configuration file for the workflow (2) to associate. The URL MUST NOT begin with a slash (/).

configVersion: A string prepended by a 'v' character identifying the **displayed version** of the workflow configuration file to use. MUST be empty or omitted, or a valid version number recognized by the server. If it is empty or omitted, the latest version of the file is used.

3.1.4.3.2.2 AssociateWorkflowMarkupResponse

The **AssociateWorkflowMarkupResponse** element MUST be returned by the server. If the operation succeeds, it returns information about the association. If the operation fails, it returns an error message that can be displayed to the user. The operation might fail for reasons including, but not limited to:

- The **configUrl** parameter points to a file that does not exist.

- The **configUrl** parameter points to an empty file.

- The **configVersion** parameter refers to a non-existent version of the workflow configuration file.

- The workflow configuration file contains incomplete or invalid data.

- The server reaches an error state while associating the workflow with the targeted object.

This element is defined as follows:

```
<s:element name="AssociateWorkflowMarkupResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="AssociateWorkflowMarkupResult" type="s:string" minOccurs="1"
maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

AssociateWorkflowMarkupResult: An XML-encoded string that, when decoded, is either one **DeclarativeWorkflowDisabled** element, one **Success** element, or one **Error** element, defined as follows:

DeclarativeWorkflowDisabled: Indicates that the operation failed because an administrator has disabled declarative workflow features on the server. MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="DeclarativeWorkflowDisabled" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema" />
```

Success: Indicates that the association operation was successful. MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="Success" type="s1:guid" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema" />
```

When decoded, this element contains a GUID that uniquely identifies the workflow (2).

Error: Indicates that the association operation failed for a reason other than disabling of the workflow (2) feature. MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="Error" type="s:string" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema"/>
```

When decoded, this element MUST contain a localized error string suitable for display to the user.

3.1.4.3.3   Complex Types

None.

3.1.4.3.4   Simple Types

None.

3.1.4.3.5   Attributes

None.

3.1.4.3.6   Groups

None.

3.1.4.3.7   Attribute Groups

None.

3.1.4.4   ConvertWebPartFormat

The **ConvertWebPartFormat** operation<22> converts a Web Part from persistence format to page markup, or vice-versa.<23>

A Web Part can exist in multiple forms. One is an implementation-specific standalone persistence format that can be used to save a Web Part as a single file, whether on the server or elsewhere. Another is the implementation-specific markup that is inserted into page markup to cause the server to render that Web Part when the user browses the page. This operation is defined as follows:

```
<wsdl:operation name="ConvertWebPartFormat" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="ConvertWebPartFormatSoapIn" />
   <wsdl:output message="ConvertWebPartFormatSoapOut" />
</wsdl:operation>
```

The client sends a **ConvertWebPartFormatSoapIn** request message, and the server responds with a **ConvertWebPartFormatSoapOut** response message.

3.1.4.4.1 Messages

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

Message	Description
ConvertWebPartFormatSoapIn	A request to initiate a **ConvertWebPartFormat** operation on the protocol server.
ConvertWebPartFormatSoapOut	A response from the protocol server at completion of the **ConvertWebPartFormat** operation.

3.1.4.4.1.1 ConvertWebPartFormatSoapIn

The **ConvertWebPartFormatSoapIn** message is the request message for **ConvertWebPartFormat**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/ConvertWebPartFormat
```

The SOAP Body contains a **ConvertWebPartFormat** element.

3.1.4.4.1.2 ConvertWebPartFormatSoapOut

The **ConvertWebPartFormatSoapOut** message is the response message for **ConvertWebPartFormat**.

The SOAP Body contains a **ConvertWebPartFormatResponse** element.

3.1.4.4.2 Elements

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

Element	Description
ConvertWebPartFormat	A request to initiate a **ConvertWebPartFormatSoapIn** operation on the protocol server.

Release: September 12, 2012

Element	Description
ConvertWebPartFormatResponse	A response from the protocol server at completion of the **ConvertWebPartFormatSoapOut** operation.

3.1.4.4.2.1 ConvertWebPartFormat

The **ConvertWebPartFormat** element is defined as follows:

```
<s:element name="ConvertWebPartFormat" xmlns:s="http://www.w3.org/2001/XMLSchema" >
   <s:complexType>
      <s:sequence>
         <s:element name="inputFormat" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="formatConversionOption" type="tns:FormatConversionOption"
minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

inputFormat: An XML-encoded Web Part markup in the input format indicated by the **formatConversionOption** element.

formatConversionOption: Specifies the format of the **inputFormat** element and the desired output format.

3.1.4.4.2.2 ConvertWebPartFormatResponse

The **ConvertWebPartFormatResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="ConvertWebPartFormatResponse" xmlns:s="http://www.w3.org/2001/XMLSchema" >
   <s:complexType>
      <s:sequence>
         <s:element name="ConvertWebPartFormatResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

ConvertWebPartFormatResult: An XML-encoded Web Part markup in the output format indicated by the **formatConversionOption** element.

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

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

Element	Description
FormatConversionOption	Enumerates the different format conversion operations. that are available.

3.1.4.4.4.1 FormatConversionOption

The **FormatConversionOption** simple type enumerates the different format conversion operations. that are available. This simple type is defined as follows:

```
<s:simpleType name="FormatConversionOption" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:restriction base="s:string">
      <s:enumeration value="ConvertToWebPartExportFormat"/>
      <s:enumeration value="ConvertToWebPartDesignerPersistenceFormat"/>
   </s:restriction>
</s:simpleType>
```

The following table specifies the possible values for this element.

Value	Description
ConvertToWebPartExportFormat	The input string is in page markup format and is converted to a standalone persistence format. When editing a page in a client application, this option is used when exporting a Web Part to its own file.
ConvertToWebPartDesignerPersistenceFormat	The input string is in a standalone persistence format and is converted to page markup. When editing a page in a client application, this option is used when importing a Web Part from a file into a page.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 DeleteWebPart

The **DeleteWebPart** operation removes a Web Part from a view of a page. This operation is defined as follows:

```
<wsdl:operation name="DeleteWebPart" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="DeleteWebPartSoapIn" />
   <wsdl:output message="DeleteWebPartSoapOut" />
</wsdl:operation>
```

The client sends a **DeleteWebPartSoapIn** request message, and the server responds with a **DeleteWebPartSoapOut** response message.

3.1.4.5.1 Messages

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

Message	Description
DeleteWebPartSoapIn	A request to initiate a **DeleteWebPart** operation on the protocol server.
DeleteWebPartSoapOut	A response from the protocol server at completion of the **DeleteWebPart** operation.

3.1.4.5.1.1 DeleteWebPartSoapIn

The **DeleteWebPartSoapIn** message is the request message for **DeleteWebPart**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/DeleteWebPart
```

The SOAP Body contains a **DeleteWebPart** element.

3.1.4.5.1.2 DeleteWebPartSoapOut

The **DeleteWebPartSoapOut** message is the response message for **DeleteWebPart**.

The SOAP Body contains a **DeleteWebPartResponse** element.

3.1.4.5.2 Elements

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

Element	Description
DeleteWebPart	A request to initiate a **DeleteWebPartSoapIn** operation on the protocol server.
DeleteWebPartResponse	A response from the protocol server at completion of the **DeleteWebPartSoapOut** operation.

3.1.4.5.2.1 DeleteWebPart

The **DeleteWebPart** element deletes a Web Part from a view of a page. This element is defined as follows:

```
<s:element name="DeleteWebPart" xmlns:s="http://www.w3.org/2001/XMLSchema">
  <s:complexType>
    <s:sequence>
      <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
```

```
            <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

pageUrl: MUST be one of the following: an absolute URL, a server-relative URL, or a site-relative URL. It specifies the location of the page. The location specified MUST be on the current site (2) or any child site (2) of the current site (2) that is also in the current site collection.

storageKey: A GUID that uniquely identifies the Web Part and specifies which Web Part to delete from the page. MUST correspond to a Web Part that is specified in a Web Part zone in the view specified by the **storage** parameter.

storage: Specifies whether the specified Web Part is removed from the public view or the current user's personal view of the page. If this is set to "None" or is unset, it is treated as if it is set to "Personal". If this is set to "Personal", the Web Part MUST exist in the current user's personal view and MUST NOT exist in the public view. If this is set to "Shared", the Web Part MUST exist in the public view and MUST NOT exist in the user's personal view.

3.1.4.5.2.2   DeleteWebPartResponse

The **DeleteWebPartResponse** element MUST be sent by the server unless there is an error. This element is defined as follows:

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

3.1.4.5.3   Complex Types

None.

3.1.4.5.4   Simple Types

None.

3.1.4.5.5   Attributes

None.

3.1.4.5.6   Groups

None.

3.1.4.5.7   Attribute Groups

None.

3.1.4.6   ExecuteProxyUpdates

The **ExecuteProxyUpdates** operation<24> transmits a modification to a page containing controls to the server so that the implementation of the page running in the server environment can

evaluate the same change. Then, the server can return the updated markup for the controls on the page. This operation is defined as follows:

```
<wsdl:operation name="ExecuteProxyUpdates" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="ExecuteProxyUpdatesSoapIn" />
    <wsdl:output message="ExecuteProxyUpdatesSoapOut" />
</wsdl:operation>
```

The client sends an **ExecuteProxyUpdatesSoapIn** request message, and the server responds with an **ExecuteProxyUpdatesSoapOut** response message.

3.1.4.6.1   Messages

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

Message	Description
ExecuteProxyUpdatesSoapIn	A request to initiate a **ExecuteProxyUpdates** operation on the protocol server.
ExecuteProxyUpdatesSoapOut	A response from the protocol server at completion of the **ExecuteProxyUpdates** operation.

3.1.4.6.1.1   ExecuteProxyUpdatesSoapIn

The **ExecuteProxyUpdatesSoapIn** message is the request message for **ExecuteProxyUpdates**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/ExecuteProxyUpdates
```

The SOAP Body contains an **ExecuteProxyUpdates** element.

3.1.4.6.1.2   ExecuteProxyUpdatesSoapOut

The **ExecuteProxyUpdatesSoapOut** message is the response message for **ExecuteProxyUpdates**.

The SOAP Body contains an **ExecuteProxyUpdatesResponse** element.

3.1.4.6.2   Elements

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

Element	Description
ExecuteProxyUpdates	A request to initiate a **ExecuteProxyUpdatesSoapIn** operation on the protocol server.
ExecuteProxyUpdatesResponse	A response from the protocol server at completion of the **ExecuteProxyUpdatesSoapOut** operation.

3.1.4.6.2.1 ExecuteProxyUpdates

The **ExecuteProxyUpdates** element indicates that a change was made to a control that requires evaluation in the server environment to produce the correct markup. This markup can be both the persisted markup on the page and display markup to render the control for editing. This element is defined as follows:

```
<s:element name="ExecuteProxyUpdates" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="updateData" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

updateData: An XML-encoded string containing details of the update to evaluate on the server. The client MUST send a value that matches this schema, as follows:

```
<s:schema elementFormDefault="qualified" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:element name="UpdateTransaction">
      <s:complexType>
         <s:sequence>
            <s:element name="Update" minOccurs="1" maxOccurs="unbounded">
               <s:complexType>
                  <s:sequence>
                     <s:element name="Document">
                        <s:complexType>
                           <s:sequence>
                              <s:element name="Register" minOccurs="0" maxOccurs="unbounded">
                                 <s:complexType>
                                    <s:sequence>
                                    </s:sequence>
                                    <s:attribute name="TagPrefix" type="s:string"
  use="required" />
                                    <s:attribute name="Src" type="s:string" use="optional" />
                                    <s:attribute name="TagName" type="s:string"
  use="optional" />
                                    <s:attribute name="Assembly" type="s:string"
  use="optional" />
                                    <s:attribute name="Namespace" type="s:string"
  use="optional" />
                                 </s:complexType>
                              </s:element>
                              <s:element name="Control" type="controlType" minOccurs="0"
  maxOccurs="unbounded">
                              </s:element>
                           </s:sequence>
                           <s:attribute name="Url" type="s:string" use="required" />
                           <s:attribute name="ContextUrl" type="s:string" use="optional" />
                           <s:attribute name="WebUrl" type="s:string" use="optional" />
                        </s:complexType>
                     </s:element>
                     <s:element name="Actions" minOccurs="0" maxOccurs="1">
                        <s:complexType>
                           <s:choice maxOccurs="unbounded">
                              <s:element name="Method">
                                 <s:complexType>
                                    <s:sequence>
```

```xml
                                                   <s:element name="Param" minOccurs="0"
maxOccurs="unbounded">
                                                       <s:complexType>
                                                          <s:simpleContent>
                                                             <s:extension base="s:string">
                                                                <s:attribute name="Type" type="s:string"
use="required"/>
                                                                <s:attribute name="Text" type="s:string"
use="optional"/>
                                                             </s:extension>
                                                          </s:simpleContent>
                                                       </s:complexType>
                                                   </s:element>
                                                </s:sequence>
                                                <s:attribute name="UpdateID" type="s:nonNegativeInteger"
use="required"/>
                                                <s:attribute name="Name" type="s:string" use="optional"
/>
                                          </s:complexType>
                                       </s:element>
                                       <s:element name="Prop" type="propType" />
                                    </s:choice>
                              </s:complexType>
                           </s:element>
                     </s:sequence>
                  </s:complexType>
               </s:element>
               <s:complexType name="controlType">
                  <s:sequence>
                     <s:element name="Region" minOccurs="0" maxOccurs="unbounded">
                        <s:complexType>
                           <s:sequence minOccurs="0" maxOccurs="unbounded">
                              <s:element name="Control" type="controlType" />
                           </s:sequence>
                           <s:attribute name="Index" type="s:nonNegativeInteger" use="required" />
                        </s:complexType>
                     </s:element>
                  </s:sequence>
                  <s:attribute name="NeedsPreview" use="optional">
                     <s:simpleType>
                        <s:restriction base="s:string">
                           <s:enumeration value="True" />
                           <s:enumeration value="False" />
                        </s:restriction>
                     </s:simpleType>
                  </s:attribute>
                  <s:attribute name="ForceUseDesigner" use="optional">
                     <s:simpleType>
```

```
            <s:restriction base="s:string">
               <s:enumeration value="True" />
               <s:enumeration value="False" />
            </s:restriction>
         </s:simpleType>
      </s:attribute>
      <s:attribute name="ID" type="s:string" use="optional" />
      <s:attribute name="TagName" type="s:string" use="required" />
      <s:attribute name="OuterHtml" type="s:string" use="required" />
      <s:attribute name="UpdateID" type="s:string" use="required" />
   </s:complexType>
   <s:complexType name="subPropType">
      <s:sequence>
         <s:element name="Prop" type="subPropType" minOccurs="0" maxOccurs="unbounded" />
      </s:sequence>
      <s:attribute name="Name" type="s:string" use="required" />
      <s:attribute name="Text" type="s:string" use="optional" />
   </s:complexType>
   <s:complexType name="propType">
      <s:sequence>
         <s:element name="Prop" type="subPropType" minOccurs="0" maxOccurs="unbounded" />
      </s:sequence>
      <s:attribute name="UpdateID" type="s:string" use="required" />
      <s:attribute name="Source" use="optional">
         <s:simpleType>
            <s:restriction base="s:string">
               <s:enumeration value="Control" />
               <s:enumeration value="Designer" />
               <s:enumeration value="Bound" />
            </s:restriction>
         </s:simpleType>
      </s:attribute>
      <s:attribute name="Name" type="s:string" use="required" />
      <s:attribute name="Text" type="s:string" use="optional" />
   </s:complexType>
</s:schema>
```

*.Prop: For properties for which the client does not have the full logic to serialize the value into the **Text** attribute, the client MUST specify child **Prop** elements to specify how the server is to serialize the sub-values that constitute the full value of the property. If there are any **Prop** child elements, the **Text** attribute MUST NOT be set.

UpdateTransaction: The top-level container for **Update** elements.

UpdateTransaction.Update: Represents an individual page, the controls on it, and the updates to make to those controls.

The server MUST accept any attributes on this element and the server MUST return them as attributes of the **UpdateResponse.Update** element that is part of the response to this request. This enables the client to pass a set of opaque data to the server to identify which response **UpdateResponse.Update** element matches which request **UpdateTransaction.Update** element.

UpdateTransaction.Update.Type: This attribute MUST be set to "Document".

UpdateTransaction.Update.Document: Specifies the content of the page necessary for performing the requested updates.

UpdateTransaction.Update.Document.Url: The URL of the page being modified.

UpdateTransaction.Update.ContextUrl: If non-empty and different from **Url**, identifies this as a nested page, where one page reads content from another page to use as portions of its markup for its browse-time processing on the server. The value of **Url** identifies the page that contains the markup, and **ContextUrl** identifies the pages that is reading content from the page at **Url**.

UpdateTransaction.Update.WebUrl: The URL of the site (2) that contains this page.

UpdateTransaction.Update.Document.Register: Specifies the **register directives** that are necessary to instantiate the controls on the page. Either the **Src** and the **TagName** attributes or the **Namespace** and the **Assembly** attributes MUST be set.

UpdateTransaction.Update.Document.Register.TagPrefix: The **tag prefix** for the register directive.

UpdateTransaction.Update.Document.Register.Src: The source for the register directive. If the client sets the **Src** attribute, the client MUST also set the **TagName** and the client MUST NOT set the **Assembly** and the **Namespace** attributes.

UpdateTransaction.Update.Document.Register.TagName: The tag name of the register directive. If the client sets the **TagName** attribute, the client MUST also set the **Src** attribute and the client MUST NOT set the **Assembly** and the **Namespace** attributes.

UpdateTransaction.Update.Document.Register.Assembly: The assembly of the register directive. If the client sets this attribute, the client MUST also set the **Namespace** attribute and the client MUST NOT set the **Src** and the **TagName** attributes.

UpdateTransaction.Update.Document.Register.Namespace: The namespace of the register directive. If the client sets the **Namespace** attribute, the client MUST also set the **Assembly** attribute and the client MUST NOT set the **Src** and the **TagName** attributes.

UpdateTransaction.Update.Document.Control: Specifies one control that exists in the document.

UpdateTransaction.Update.Actions: The container for the list of updates to make to the page.

UpdateTransaction.Update.Actions.Method: Specifies a single update operation to perform on a single control on the server.

UpdateTransaction.Update.Actions.Method.UpdateID: A number that maps to the **ID** attribute of an **UpdateTransaction.Update.Document.*.Control** element to determine on which control the operation is performed.

UpdateTransaction.Update.Actions.Method.Name: The name of the operation to perform. If not set, the parent **Method** element is ignored.

UpdateTransaction.Update.Actions.Method.Param: For operations that take arguments, each **Param** element specifies one argument of the operation.

UpdateTransaction.Update.Actions.Method.Param.Type: Specifies the class of the object used for the parameter of the method.

UpdateTransaction.Update.Actions.Method.Param.Text: Specifies the value of the object used for the parameter of the method. The server MUST decode the string and process the request.

subPropType: Specifies the serialized value of an object contained in the ancestor **propType** object.

subPropType.Name: The name of the attribute to update.

subPropType.Text: The full serialized value of the object. If the client sets this attribute, the client MUST NOT create any **Prop** sub elements.

controlType: Specifies a single control in the document.

controlType.NeedsPreview: If set to TRUE, this specifies that the server MUST return new markup for display in a page editing tool. If set to FALSE, this specifies that the server MUST NOT return new markup for display in a page editing tool.

controlType.ForceUseDesigner: If set to TRUE, this specifies that the server MUST use the object type of the control to retrieve a name for an auxiliary object type. The auxiliary object type MUST be used to make the updates to the control.

controlType.ID: The identifier of the control that uniquely identifies it on a page.

controlType.TagName: The implementation-specific<25> name for the object-oriented type that is the control.

controlType.OuterHtml: The persisted markup of the control.

controlType.UpdateID: A number that the client sets that the server MUST return as part of updates to this control.

controlType.Region: Specifies a region in a control that can contain other controls.

controlType.Region.Control: Specifies one of the controls located in the parent **controlType.Region** element region.

controlType.Region.Index: The numerical index of a region in a control that can contain other controls.

propType: Specifies an update to a single sub-object of the object-oriented type that is the control. It MUST be serializable to text. However, the client does not necessarily contain the logic to serialize the value. If the client does not contain the logic to serialize the value, it MUST break the object into its sub-objects to have the server set the value of the sub-object.

propType.UpdateID: Matches the **UpdateTransaction.Update.Document.*.Control.UpdateID** of the control to update.

propType.Name: The name of the sub-object to update.

propType.Source: Specifies how the server MUST use the provided value. MUST be set to one of the values in the following table.

Value	Description
Control	The server MUST set the value directly on the control.
Bound	If the **Text** property is set, the server MUST split the value of the **Text** property on the colon (:). The portion before the first colon is the identifier of a control from which to obtain the value. The portion after the first colon is the name of the sub-object of the control to which the value binds.
Designer	The server MUST use an auxiliary type on the server to set the value of the control.

propType.Text: The full serialized value of the object. If the client sets this attribute, the client MUST NOT create any **Prop** sub-elements.

3.1.4.6.2.2 ExecuteProxyUpdatesResponse

The **ExecuteProxyUpdatesResponse** element is the response to the **ExecuteProxyUpdates** request from the client. This element is defined as follows:

```
<s:element name="ExecuteProxyUpdatesResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
        <s:sequence>
            <s:element name="ExecuteProxyUpdatesResult" type="s:string" minOccurs="1"
maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
```

ExecuteProxyUpdatesResult: This response contains the updated markup and previews of the Web controls as specified in the request. The server MUST build the result matching this schema. The server MUST then **XML encode** the resulting XML, and MUST then return it as a string to the client.

```
<s:schema elementFormDefault="qualified" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType name="PType">
        <s:sequence>
            <s:choice maxOccurs="unbounded">
                <s:element name="P" type="PType" minOccurs="0" maxOccurs="unbounded" />
            </s:choice>
        </s:sequence>
        <s:attribute name="ID" type="s:nonNegativeInteger" use="optional" />
        <s:attribute name="N" type="s:string" use="required" />
        <s:attribute name="Y" type="s:string" use="optional" />
        <s:attribute name="R" type="s:string" use="optional" />
        <s:attribute name="E" type="s:string" use="optional" />
        <s:attribute name="Serial" type="s:string" use="optional" />
        <s:attribute name="T" type="s:string" use="optional" />
        <s:attribute name="Bound" use="optional">
            <s:simpleType>
                <s:restriction base="s:string">
                    <s:enumeration value="True" />
                </s:restriction>
            </s:simpleType>
        </s:attribute>
        <s:attribute name="AssemblyQualifiedName" type="s:string" use="optional" />
    </s:complexType>
    <s:element name="UpdateResponse">
        <s:complexType>
            <s:sequence>
                <s:element name="Update" minOccurs="1" maxOccurs="unbounded">
                    <s:complexType>
                        <s:choice maxOccurs="unbounded">
                            <s:element name="Method">
                                <s:complexType>
                                    <s:sequence>
                                        <s:element name="Param">
                                            <s:complexType>
                                                <s:sequence>
```

```
                            </s:sequence>
                            <s:attribute name="Y" type="s:string" use="required"/>
                            <s:attribute name="T" type="s:string" use="required"/>
                        </s:complexType>
                    </s:element>
                </s:sequence>
                <s:attribute name="UpdateID" type="s:nonNegativeInteger"
use="required" />
                <s:attribute name="N" type="s:string" use="optional" />
            </s:complexType>
        </s:element>
        <s:element name="Values">
            <s:complexType>
                <s:sequence>
                    <s:element name="P" type="PType" minOccurs="0"
maxOccurs="unbounded"/>
                </s:sequence>
                <s:attribute name="UpdateID" type="s:nonNegativeInteger"
use="required" />
            </s:complexType>
        </s:element>
        <s:element name="Templates">
            <s:complexType>
                <s:sequence>
                    <s:element name="Group" minOccurs="0" maxOccurs="unbounded">
                        <s:complexType>
                            <s:sequence>
                                <s:element name="Template" minOccurs="0"
maxOccurs="unbounded">
                                    <s:complexType>
                                        <s:sequence>
                                        </s:sequence>
                                        <s:attribute name="Name" type="s:string"
use="required" />
                                        <s:attribute name="Prop" type="s:string"
use="optional" />
                                        <s:attribute name="Flags" use="optional">
                                            <s:simpleType>
                                                <s:restriction base="s:string">
                                                    <s:pattern value="S?D?" />
                                                </s:restriction>
                                            </s:simpleType>
                                        </s:attribute>
                                        <s:attribute name="Content" type="s:string"
use="optional" />
                                    </s:complexType>
                                </s:element>
                            </s:sequence>
                            <s:attribute name="Name" type="s:string" use="required"
/>
                        </s:complexType>
                    </s:element>
                </s:sequence>
                <s:attribute name="UpdateID" type="s:nonNegativeInteger"
use="required"/>
            </s:complexType>
        </s:element>
    </s:choice>
    <s:anyAttribute namespace="##any" processContents="skip" />
</s:complexType>
```

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

PType: Specifies a sub-object of the control and the value to which to change the sub-object. This will be called Referred Attribute (RA) for the remainder of this section.

PType.P: Specifies a sub object of the Referred Attribute if the RA is composed as a collection of sub-properties.

PType.ID: Used in conjunction with the **PType.R** attribute. If this attribute is set, a sibling **PType** element MAY set the value of its **R** attribute to the value of this attribute, instead of fully serializing the value to which to change the attribute. This can achieve a performance improvement by reducing the total number of bytes transferred.

PType.N: The name of the RA to change.

PType.Y: The object type of the value to which to change the RA. If the type is an enumeration, the **PType.E** attribute is set on this element. If the type is the name of a type, the **AssemblyQualifiedName** attribute is set on this element.

PType.R: If set, the value to which to change the RA has already been specified by a previous sibling **PType** element. If the value of **PType.R** is set to -1, it refers to the value of an empty string. If the value of **PType.R** is set to zero, it refers to the control that is being modified. Otherwise, the value of **R** is the value of the **ID** of the previous sibling that has the same value.

PType.E: If set, this specifies the value of the enumeration type to which to set the RA. If set, **T**, **Serial**, **Bound**, and **AssemblyQualifiedName** MUST NOT be set.

PType.Serial: A base 64-encoded serialized form of the object to which to set the attribute. If set, **T**, **E**, **Bound**, and **AssemblyQualifiedName** MUST NOT be set.

PType.T: A textual serialized object to which to set the RA. If set, **E**, **AssemblyQualifiedName**, and **Serial** MUST NOT be set.

PType.Bound: If set to TRUE, specifies that the runtime value of the RA is specified by another attribute on another control. If **Bound** is set on this element, **T** MUST be set on this element. If **Bound** is set on this element, **E**, **AssemblyQualifiedName**, and **Serial** MUST NOT be set.

PType.AssemblyQualifiedName: If the value of the RA is the name of a type in an object-oriented system, this attribute is the name of that type. If set, **T**, **E**, **Bound**, and **Serial** MUST NOT be set.

UpdateResponse: The root element that contains all **Update** sub-elements.

UpdateResponse.Update: The attributes present on this element MUST match the **UpdateTransaction.Update** element from the request. This enables the client to match a response with a request.

UpdateResponse.Update.Method: Specifies a single method for the client to evaluate to be in the same state as the server-evaluated page.

UpdateResponse.Update.Method.UpdateID: Identifies which control to evaluate the method on.

UpdateResponse.Update.Method.N: The name of the method to evaluate. The exact details of each method are implementation-specific.<u>[26]</u>

UpdateResponse.Update.Method.Param: Specifies one argument of the method.

UpdateResponse.Update.Method.Param.Y: Specifies the object type of the argument.

UpdateResponse.Update.Method.Param.T: Specifies the object value of the argument.

UpdateResponse.Update.Values: Specifies a set of values to change on a control on the page.

UpdateResponse.Update.Values.UpdateID: Identifies which control to change the value on.

UpdateResponse.Update.Templates: Specifies an update to the sets of nested controls in a control.

UpdateResponse.Update.Templates.UpdateID: Identifies which control to change the value on.

UpdateResponse.Update.Templates.Group: Specifies updates to one set of nested controls in a control.

UpdateResponse.Update.Templates.Group.Name: The name of the set of controls.

UpdateResponse.Update.Templates.Group.Template: Each **Template** element specifies changes made to one nesting of controls in a control.

UpdateResponse.Update.Templates.Group.Template.Name: The name of the nested control to update.

UpdateResponse.Update.Templates.Group.Template.Prop: If not set, use the same value as **Name**. The **Prop** attribute specifies the nested control to update.

UpdateResponse.Update.Templates.Group.Template.Flags: Contains one or more of the characters in the following table to specify additional processing instructions.

Character	Description
S	If present, this region supports only content that directly maps to controls evaluated on the server.
D	If present, this region supports data binding to the contents of other regions.

UpdateResponse.Update.Templates.Group.Template.Content: The new content of the nested control.

3.1.4.6.3   Complex Types

None.

3.1.4.6.4   Simple Types

None.

3.1.4.6.5   Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 FetchLegalWorkflowActions

The **FetchLegalWorkflowActions** operation<27> returns localized UI strings and specifications for the actions (2) and conditions available for use in a declarative workflow on the server. This operation is defined as follows:

```
<wsdl:operation name="FetchLegalWorkflowActions"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="FetchLegalWorkflowActionsSoapIn" />
   <wsdl:output message="FetchLegalWorkflowActionsSoapOut" />
</wsdl:operation>
```

The client sends a **FetchLegalWorkflowActionsSoapIn** request message, and the server responds with a **FetchLegalWorkflowActionsSoapOut** response message.

3.1.4.7.1 Messages

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

Message	Description
FetchLegalWorkflowActionsSoapIn	A request to initiate a **FetchLegalWorkflowActions** operation on the protocol server.
FetchLegalWorkflowActionsSoapOut	A response from the protocol server at completion of the **FetchLegalWorkflowActions** operation.

3.1.4.7.1.1 FetchLegalWorkflowActionsSoapIn

The **FetchLegalWorkflowActionsSoapIn**message is the request message for **FetchLegalWorkflowActions**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/FetchLegalWorkflowActions
```

The SOAP Body contains a **FetchLegalWorkflowActions** element.

3.1.4.7.1.2 FetchLegalWorkflowActionsSoapOut

The **FetchLegalWorkflowActionsSoapOut** message is the response message for **FetchLegalWorkflowActions**.

The SOAP Body contains a **FetchLegalWorkflowActionsResponse** element.

3.1.4.7.2 Elements

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

Element	Description
FetchLegalWorkflowActions	A request to initiate a **FetchLegalWorkflowActionsSoapIn** operation on the protocol server.
FetchLegalWorkflowActionsResponse	A response from the protocol server at completion of the **FetchLegalWorkflowActionSoapOut** operation.

3.1.4.7.2.1 FetchLegalWorkflowActions

The **FetchLegalWorkflowActions** element is defined as follows:

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

3.1.4.7.2.2 FetchLegalWorkflowActionsResponse

The **FetchLegalWorkflowActionsResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="FetchLegalWorkflowActionsResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="FetchLegalWorkflowActionsResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

FetchLegalWorkflowActionsResult: An XML-encoded string that, when decoded, is either one **DeclarativeWorkflowDisabled** element or one **WorkflowInfo** element, defined as follows:

DeclarativeWorkflowDisabled: Indicates that the operation failed because an administrator has disabled declarative workflow features on the server. When this element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="DeclarativeWorkflowDisabled" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema"/>
```

WorkflowInfo: A root element containing information about the actions (2) and conditions that are available for use in workflows (2) on the server.

```
<s:schema xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:core="http://schemas.microsoft.com/sharepoint/soap/">
<s:import namespace="http://schemas.microsoft.com/sharepoint/soap/" />
<s:element name="WorkflowInfo">
```

```
<s:complexType>
    <s:all>
        <s:element name="Conditions" minOccurs="1" maxOccurs="1">
            <s:complexType>
                <s:sequence>
                    <s:element name="Default" minOccurs="0" maxOccurs="1">
                        <s:complexType>
                            <s:sequence>
                                <s:element name="RuleDesigner" type="ruleDesignerType" />
                            </s:sequence>
                            <s:attribute name="ItemKeyType" type="s:string" />
                        </s:complexType>
                    </s:element>
                    <s:element name="Condition" minOccurs="0" maxOccurs="unbounded">
                        <s:complexType>
                            <s:all>
                                <s:element name="RuleDesigner" type="ruleDesignerType"
 minOccurs="1" maxOccurs="1" />
                                <s:element name="Parameters" type="parametersType" minOccurs="1"
 maxOccurs="1" />
                            </s:all>
                            <s:attribute name="Type">
                                <s:simpleType>
                                    <s:restriction base="s:string">
                                        <s:enumeration value="Advanced" />
                                    </s:restriction>
                                </s:simpleType>
                            </s:attribute>
                            <s:attribute name="Name" type="s:string" use="required" />
                            <s:attribute name="AppliesTo" type="appliesToTypes" use="required" />
                            <s:attribute name="UsesCurrentItem" type="s:boolean" default="false"
 />
                            <s:attribute name="ItemKeyType" type="s:string" />
                            <s:attribute name="FunctionName" type="s:string" />
                            <s:attribute name="Assembly" type="s:string" />
                            <s:attribute name="ClassName" type="s:string" />
                            <s:attributeGroup ref="invocationTagAttributeGroup" />
                        </s:complexType>
                    </s:element>
                </s:sequence>
                <s:attribute name="And" type="s:string" use="required" />
                <s:attribute name="Or" type="s:string" use="required" />
                <s:attribute name="When" type="s:string" use="required" />
                <s:attribute name="Else" type="s:string" use="required" />
                <s:attribute name="Not" type="s:string" use="required" />
            </s:complexType>
        </s:element>
        <s:element name="Actions" minOccurs="1" maxOccurs="1">
            <s:complexType>
                <s:sequence>
                    <s:element name="Default" type="defaultElementType" minOccurs="0"
 maxOccurs="1" />
                    <s:element name="Action" minOccurs="0" maxOccurs="unbounded">
                        <s:complexType>
                            <s:all>
                                <s:element name="Parameters" type="parametersType" minOccurs="0"
 maxOccurs="1" />
                                <s:element name="RuleDesigner" type="ruleDesignerType"
 minOccurs="1" maxOccurs="1" />
```

```
                              <s:element name="DataSources" type="dataSourcesType" minOccurs="0"
   maxOccurs="1" />
                              <s:element name="Modifications" minOccurs="0" maxOccurs="1">
                                 <s:complexType>
                                    <s:sequence>
                                       <s:element name="Modification" minOccurs="1"
   maxOccurs="unbounded">
                                          <s:complexType>
                                             <s:sequence />
                                             <s:attribute name="TypeId" type="s:string"
   use="required" />
                                             <s:attribute name="NameFormat" use="required">
                                                <s:simpleType>
                                                   <s:restriction base="s:string">
                                                      <s:pattern value="([^\{]|\\\{|(\{0\}))*"
   />
                                                   </s:restriction>
                                                </s:simpleType>
                                             </s:attribute>
                                             <s:attribute name="FormURN" type="s:string"
   use="required" />
                                          </s:complexType>
                                       </s:element>
                                    </s:sequence>
                                 </s:complexType>
                              </s:element>
                              <s:element name="ActionVariables" minOccurs="0" maxOccurs="1">
                                 <s:complexType>
                                    <s:sequence>
                                       <s:any namespace="##any" minOccurs="0"
   maxOccurs="unbounded" processContents="skip" />
                                    </s:sequence>
                                 </s:complexType>
                              </s:element>
                              <s:element name="ActionBody" minOccurs="0" maxOccurs="1">
                                 <s:complexType>
                                    <s:sequence>
                                       <s:any namespace="##any" minOccurs="0"
   maxOccurs="unbounded" processContents="skip" />
                                    </s:sequence>
                                 </s:complexType>
                              </s:element>
                              <s:element name="ActionConditions" minOccurs="0" maxOccurs="1">
                                 <s:complexType>
                                    <s:sequence>
                                       <s:any namespace="##any" minOccurs="0"
   maxOccurs="unbounded" processContents="skip" />
                                    </s:sequence>
                                 </s:complexType>
                              </s:element>
                           </s:all>
                           <s:attribute name="Name" type="s:string" use="required" />
                           <s:attribute name="ClassName" type="s:string" use="required" />
                           <s:attribute name="Assembly" type="s:string" use="required" />
                           <s:attribute name="FunctionName" type="s:string" default=""/>
                           <s:attribute name="Category" type="s:string" />
                           <s:attribute name="CreatesTask" type="s:boolean" default="false" />
                           <s:attribute name="CreatesInList" type="s:string" />
                           <s:attribute name="AppliesTo" type="appliesToTypes" use="required" />
                           <s:attribute name="IsError" type="s:boolean" default="false" />
```

```xml
                        <s:attribute name="ListModeration" type="s:boolean" default="false"
  />
                        <s:attribute name="UsesCurrentItem" type="s:boolean" default="false"
  />
                        <s:attributeGroup ref="invocationTagAttributeGroup"/>
                        <s:attribute name="CreatedTaskFormType">
                           <s:simpleType>
                              <s:restriction base="s:string">
                                 <s:enumeration value="DataCollectTask" />
                                 <s:enumeration value="GroupAssignedTask" />
                                 <s:enumeration value="TodoItemTask" />
                                 <s:enumeration value="TaskProcess" />
                              </s:restriction>
                           </s:simpleType>
                        </s:attribute>
                        <s:attribute name="__SolutionId" type="s:string" />
                        <s:attribute name="SandboxedFunction" type="s:boolean"
default="false" />
                  </s:complexType>
               </s:element>
            </s:sequence>
            <s:attribute name="Sequential" type="s:string" use="required" />
            <s:attribute name="Parallel" type="s:string" use="required" />
         </s:complexType>
         <s:unique name="eachActionHasUniqueAssemblyAndClassName">
            <s:selector xpath="./Action"/>
            <s:field xpath="@Assembly"/>
            <s:field xpath="@ClassName"/>
            <s:field xpath="@FunctionName"/>
         </s:unique>
      </s:element>
      <s:element name="CompositeSteps" minOccurs="1" maxOccurs="1">
         <s:complexType>
            <s:sequence minOccurs="0" maxOccurs="unbounded">
               <s:element name="CompositeStep">
                  <s:complexType>
                     <s:sequence>
                         <s:element name="DataSources" type="dataSourcesType"
minOccurs="0" maxOccurs="1" />
                     </s:sequence>
                     <s:attribute name="Name" type="s:string" use="required" />
                     <s:attribute name="ClassName" type="s:string" use="required" />
                     <s:attribute name="Assembly" type="s:string" use="required" />
                     <s:attribute name="AppliesTo" type="appliesToTypes" use="required" />
                     <s:attributeGroup ref="invocationTagAttributeGroup"/>
                     <s:attribute name="Order" type="s:nonNegativeInteger" default="0" />
                     <s:attribute name="Description" type="s:string" />
                  </s:complexType>
               </s:element>
            </s:sequence>
         </s:complexType>
      </s:element>
      <s:element name="VariableTypes" minOccurs="1" maxOccurs="1">
         <s:complexType>
            <s:sequence minOccurs="0" maxOccurs="unbounded">
               <s:element name="VariableType">
                  <s:complexType>
                     <s:attribute name="Type" type="s:string" use="required" />
                     <s:attribute name="DisplayName" type="s:string" use="required" />
```

66 / 224

```
                        <s:attribute name="Assembly" type="s:string" />
                        <s:attribute name="CoercionActivity" type="s:string" use="required"
   />
                        <s:attribute name="Default" type="s:boolean" default="false" />
                        <s:attribute name="ItemIdType" type="s:boolean" default="false" />
                        <s:attribute name="Hidden" type="s:boolean" default="false" />
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="AssemblyRedirects" minOccurs="1" maxOccurs="1">
        <s:complexType>
            <s:sequence   minOccurs="0" maxOccurs="unbounded">
                <s:element name="AssemblyRedirect">
                    <s:complexType>
                        <s:attribute name="From" type="s:string" use="required" />
                        <s:attribute name="To" type="s:string" use="required" />
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:element name="Coercions" minOccurs="1" maxOccurs="1">
        <s:complexType>
            <s:sequence minOccurs="0" maxOccurs="unbounded">
                <s:element name="Coercion">
                    <s:complexType>
                        <s:attribute name="Name" type="s:string" use="required" />
                        <s:attribute name="Assembly" type="s:string" use="required" />
                        <s:attribute name="DisplayName" type="s:string" use="required" />
                        <s:attribute name="CatchAll" type="s:boolean" use="required" />
                        <s:attribute name="RankDefault" type="s:integer" use="required" />
                        <s:attribute name="SourceType" type="s:string" />
                        <s:attribute name="SourceFieldType" type="core:FieldInternalType" />
                        <s:attribute name="DestinationType" type="s:string" use="required" />
                        <s:attribute name="DestinationFieldType"
   type="core:FieldInternalType" />
                        <s:attribute name="ClassName" type="s:string" />
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
    </s:all>
    <s:attribute name="Language" type="s:string" />
    </s:complexType>
</s:element>
<s:complexType name="ruleDesignerType">
    <s:sequence>
        <s:element name="FieldBind" minOccurs="0" maxOccurs="unbounded">
            <s:complexType>
                <s:choice>
                    <s:sequence>
                        <s:element name="Option" minOccurs="0" maxOccurs="unbounded">
                            <s:complexType>
                                <s:sequence>
                                </s:sequence>
                                <s:attribute name="Name" type="s:string" use="required" />
```

67 / 224

```
                        <s:attribute name="Value" type="s:string" use="required" />
                        <s:attribute name="TypeFilter" type="s:string" />
                        <s:attribute name="UnaryHides" type="s:string" />
                        </s:complexType>
                    </s:element>
                </s:sequence>
                <s:element name="DataSourceRef" minOccurs="1" maxOccurs="unbounded">
                    <s:complexType>
                        <s:sequence />
                        <s:attribute name="Ref" type="s:string" />
                    </s:complexType>
                </s:element>
            </s:choice>
            <s:attribute name="Field" type="s:string" />
            <s:attribute name="Text" type="s:string" />
            <s:attribute name="Id" type="s:positiveInteger" use="required" />
            <s:attribute name="DesignerType" use="required">
                <s:simpleType>
                    <s:restriction base="s:string">
                        <s:enumeration value="Bool" />
                        <s:enumeration value="CheckPermission" />
                        <s:enumeration value="Assignment" />
                        <s:enumeration value="TaskSummary" />
                        <s:enumeration value="ReplacePermission" />
                        <s:enumeration value="AddPermission" />
                        <s:enumeration value="RemovePermission" />
                        <s:enumeration value="StatusDropdown" />
                        <s:enumeration value="DataSourceFieldNames" />
                        <s:enumeration value="DataSourceValues" />
                        <s:enumeration value="ChooseDoclibItem" />
                        <s:enumeration value="ChooseListItem" />
                        <s:enumeration value="CreateListItem" />
                        <s:enumeration value="Date" />
                        <s:enumeration value="Dependent" />
                        <s:enumeration value="Dropdown" />
                        <s:enumeration value="Email" />
                        <s:enumeration value="FieldNames" />
                        <s:enumeration value="Float" />
                        <s:enumeration value="Hyperlink" />
                        <s:enumeration value="Integer" />
                        <s:enumeration value="ListNames" />
                        <s:enumeration value="Operator" />
                        <s:enumeration value="ParameterNames" />
                        <s:enumeration value="Person" />
                        <s:enumeration value="SinglePerson" />
                        <s:enumeration value="StringBuilder" />
                        <s:enumeration value="Survey" />
                        <s:enumeration value="Text" />
                        <s:enumeration value="TextBox" />
                        <s:enumeration value="TextArea" />
                        <s:enumeration value="UpdateListItem" />
                        <s:enumeration value="WritableFieldNames" />
                    </s:restriction>
                </s:simpleType>
            </s:attribute>
            <s:attribute name="OperatorTypeFrom" type="s:string" />
            <s:attribute name="TypeFrom" type="s:string" />
            <s:attribute name="Function" type="s:boolean" default="false" />
            <s:attribute name="Value" type="s:string" />
```

68 / 224

```
                  </s:complexType>
               </s:element>
            </s:sequence>
            <s:attribute name="Sentence" type="s:string" />
         </s:complexType>
         <s:complexType name="defaultElementType">
            <s:sequence>
               <s:element name="RuleDesigner" type="ruleDesignerType" />
            </s:sequence>
         </s:complexType>
         <s:complexType name="parametersType">
            <s:sequence>
               <s:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
                  <s:complexType>
                     <s:sequence>
                     </s:sequence>
                     <s:attribute name="Name" type="s:string" use="required" />
                     <s:attribute name="Type" type="s:string" use="required" />
                     <s:attribute name="Direction" use="required">
                        <s:simpleType>
                           <s:restriction base="s:string">
                              <s:enumeration value="In" />
                              <s:enumeration value="Out" />
                              <s:enumeration value="Optional" />
                           </s:restriction>
                        </s:simpleType>
                     </s:attribute>
                     <s:attribute name="Description" type="s:string" />
                     <s:attribute name="DesignerType">
                        <s:simpleType>
                           <s:restriction base="s:string">
                              <s:enumeration value="ContentType" />
                              <s:enumeration value="Date" />
                              <s:enumeration value="Dependent" />
                              <s:enumeration value="Dropdown" />
                              <s:enumeration value="FieldNames" />
                              <s:enumeration value="Hide" />
                              <s:enumeration value="Integer" />
                              <s:enumeration value="ItemProperties" />
                              <s:enumeration value="ListItem" />
                              <s:enumeration value="ListNames" />
                              <s:enumeration value="ParameterNames" />
                              <s:enumeration value="Person" />
                              <s:enumeration value="SinglePerson" />
                              <s:enumeration value="StatusDropdown" />
                              <s:enumeration value="StringBuilder" />
                              <s:enumeration value="TextArea" />
                              <s:enumeration value="TextBox" />
                              <s:enumeration value="WritableFieldNames" />
                           </s:restriction>
                        </s:simpleType>
                     </s:attribute>
                     <s:attribute name="ShowLookupForDropdown" type="s:boolean" default="false" />
                     <s:attribute name="DisplayName" type="s:string" />
                     <s:attribute name="InitialValue" type="s:string" />
                     <s:attribute name="InitialBinding" type="s:string" />
                  </s:complexType>
               </s:element>
            </s:sequence>
```

```
        </s:complexType>
        <s:complexType name="dataSourcesType">
            <s:sequence>
                <s:element name="DataSource" type="dataSourceType" minOccurs="0" maxOccurs="unbounded"
/>
            </s:sequence>
        </s:complexType>
        <s:complexType name="dataSourceType">
            <s:sequence>
                <s:element name="SchemaSource">
                    <s:complexType>
                        <s:choice>
                            <s:element name="Fields" minOccurs="0" maxOccurs="1">
                                <s:complexType>
                                    <s:sequence>
                                        <s:element name="Field" minOccurs="1" maxOccurs="unbounded">
                                            <s:complexType>
                                                <s:attribute name="Name" type="s:string" />
                                                <s:attribute name="DisplayName" type="s:string" />
                                                <s:attribute name="Type" type="s:string" />
                                            </s:complexType>
                                        </s:element>
                                    </s:sequence>
                                </s:complexType>
                            </s:element>
                            <s:element name="ContentType" minOccurs="0" maxOccurs="1">
                                <s:complexType>
                                    <s:attribute name="IDFrom" type="s:string" />
                                </s:complexType>
                            </s:element>
                            <s:element name="Dictionary" minOccurs="0" maxOccurs="1">
                                <s:complexType>
                                    <s:attribute name="ValueFrom" type="s:string" />
                                </s:complexType>
                            </s:element>
                            <s:element name="NestedInitiationFieldNodes" minOccurs="0" maxOccurs="1" />
                        </s:choice>
                    </s:complexType>
                </s:element>
                <s:element name="Evaluation">
                    <s:complexType>
                        <s:choice>
                            <s:element name="ActivitySource">
                                <s:complexType>
                                </s:complexType>
                            </s:element>
                            <s:element name="HashtableSource">
                                <s:complexType>
                                    <s:attribute name="HashtableFrom" type="s:string" />
                                </s:complexType>
                            </s:element>
                        </s:choice>
                    </s:complexType>
                </s:element>
            </s:sequence>
            <s:attribute name="InternalName" type="s:string" use="required" />
            <s:attribute name="Name" type="s:string" use="required" />
            <s:attribute name="NameProperty" type="s:string" use="required" />
        </s:complexType>
```

```
<s:simpleType name="appliesToTypes">
   <s:restriction base="s:string">
      <s:enumeration value="none" />
      <s:enumeration value="list" />
      <s:enumeration value="doclib" />
      <s:enumeration value="all" />
      <s:enumeration value="displayonly" />
   </s:restriction>
</s:simpleType>
<s:attributeGroup name="invocationTagAttributeGroup">
   <s:attribute name="RequiredInvocationTags" type="s:string" />
   <s:attribute name="InvocationTags" type="s:string" />
   <s:attribute name="ProhibitedInvocationTags" type="s:string" />
</s:attributeGroup>
</s:schema>
```

WorkflowInfo.Language: A language tag in the format specified in [RFC3066] that indicates the language and culture of the UI strings in the **WorkflowInfo** document.

WorkflowInfo.Conditions: Each attribute of the **Conditions** element specifies a string that SHOULD be used in the UI of the client application to represent various logical constructs. The **Conditions** element also contains a number of **Condition** elements.

WorkflowInfo.Conditions.And: A localized string prefix (in the language and culture specified by **WorkflowInfo.Language**) added to a conditional sentence to imply that the current and previous condition MUST be set to TRUE for the contained actions (2) to be processed. For example, if this string is "and", a condition with the sentence "the user approved the document" would be displayed as "and the user approved the document".

WorkflowInfo.Conditions.Or: A localized prefix (in the language and culture specified by **WorkflowInfo.Language**) added to a conditional sentence to imply that the current or previous condition MUST be set to TRUE for the contained actions (2) to be processed. For example, if this string is "or", a condition with the sentence "the user approved the document" would be displayed as "or the user approved the document".

WorkflowInfo.Conditions.When: A localized string prefix (in the language and culture specified by **WorkflowInfo.Language**) added to the sentence of the first condition in a group of one or more conditions. For example, if this string is "If", a condition with the sentence "the user approved the document" would be displayed as "If the user approved the document".

WorkflowInfo.Conditions.Else: A localized string prefix (in the language and culture specified by **WorkflowInfo.Language**) added to the sentence of the first condition in a group of one or more conditions that is only evaluated if a preceding group of conditions evaluate to FALSE. For example, if this string is "Else if", a condition with the sentence "the user approved the document" would be displayed as "Else if the user approved the document".

WorkflowInfo.Conditions.Not: Reserved for future use and MUST be ignored by the client. The server SHOULD return a localized string in the language and culture specified by **WorkflowInfo.Language** indicating the negation of logical statement. For example, a potential value in the language/culture "en-us" is "not".

WorkflowInfo.Conditions.Default: Specifies information that the client application MUST use when displaying a workflow condition that does not match any of the **Condition** elements found in the **WorkflowInfo** document returned by this operation.

WorkflowInfo.Conditions.Default.RuleDesigner: Specifies the sentence that the client application MUST use when displaying a workflow condition that does not match any of the **Condition** elements found in the **WorkflowInfo** document returned by this operation.

WorkflowInfo.Conditions.Default.ItemKeyType: Specifies the implementation specific<28> identifier for the default object type in an object oriented system used to store the identity of a list item in a list (1) that the client application MUST use when displaying a workflow condition that does not match any of the **Condition** elements found in the **WorkflowInfo** document returned by this operation.

WorkflowInfo.Conditions.Condition: Specifies the UI characteristics of a single condition that can be used for declarative workflows on the server.

WorkflowInfo.Conditions.Condition.Type: Specifies additional characteristics of this condition, if applicable. Possible values are described in the following table.

Value	Description
(omitted)	This condition is a standard condition.
Advanced	This condition is a generic or advanced condition that supports comparing any value to any other value of an appropriate type. There MUST NOT be more than one **Condition** element with a **Type** attribute set to "Advanced" in the document.

WorkflowInfo.Conditions.Condition.Name: A localized name for the condition suitable for display in the client UI. This name SHOULD be relatively short for display in a list of available conditions, in contrast to **ruleDesignerType.Sentence**, which is meant for display in situations that are less space-constrained. It MAY include the string "%1" that indicates that the client MUST replace it with the name of the list (1) with which the workflow(2) is associated, if the workflow(2) is associated with a list (1).

WorkflowInfo.Conditions.Condition.AppliesTo: Specifies whether the condition can be used in workflows associated with specific types of lists (1). For a description of possible values, see **appliesToTypes**.

WorkflowInfo.Conditions.Condition.UsesCurrentItem: Specifies whether the condition requires data from an item on which the workflow(2) is currently operating, and thus can only be used in workflows(2) in which there is the concept of a current item. The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Description
(omitted) false	This condition does not require a current item and can be used in any workflow(2).
true	This condition requires a current item and cannot be used in workflows(2) that do not have one.

WorkflowInfo.Conditions.Condition.FunctionName: An implementation-specific<29> string that identifies a function that can be evaluated at runtime to determine whether this condition is met.

WorkflowInfo.Conditions.Condition.Assembly: An implementation-specific<30> string that uniquely identifies the binary file that contains the function specified in FunctionName.

WorkflowInfo.Conditions.Condition.ClassName: An implementation-specific<31> string that, in conjunction with the **Assembly** attribute, uniquely identifies the class in the binary file specified by the **Assembly** attribute that contains the function specified by the **FunctionName** attribute.

WorkflowInfo.Conditions.Condition.InvocationTagGroupRef: Specifies possible nesting of this **Condition** with itself and parent **Action** and **CompositeStep** elements.

WorkflowInfo.Conditions.Condition.RuleDesigner: Contains information for displaying this condition in the client application UI. See **ruleDesignerType**.

WorkflowInfo.Conditions.Condition.Parameters: Contains information about the parameters accepted by this condition. See **parametersType**.

WorkflowInfo.Conditions.Condition.ItemKeyType: Specifies the implementation specific<32> identifier for the object type in an object oriented system to use to store the identity of a **list item** in a list (1).

WorkflowInfo.Actions: Each attribute of the **Actions** element specifies a string that SHOULD be used in the client application UI to represent various logical constructs. The **Actions** element also contains a number of **Action** elements.

WorkflowInfo.Actions.Sequential: A localized string (in the language and culture specified by **WorkflowInfo.Language**) for the client to prepend to the sentence of every action (2) except the first in a group of actions (2) that run sequentially. For example, if this string is "then", an action (2) with the sentence "perform this action" is displayed as "then perform this action".

WorkflowInfo.Actions.Parallel: A localized string (in the language and culture specified by **WorkflowInfo.Language**) for the client to prepend to the sentence of every action (2) except the first in a group of actions (2) that run simultaneously. For example, if this string is "and", an action (2) with the sentence "perform this action" is displayed as "and perform this action".

WorkflowInfo.Actions.Default: Specifies the sentence that the client application MUST use when displaying a workflow action (2) that does not match any of the **Action** elements found in the **WorkflowInfo** document returned by this operation. See **defaultElementType**.

WorkflowInfo.Actions.Action: Specifies the UI characteristics of a single action (2) that can be used for declarative workflows on the server.

WorkflowInfo.Actions.Action.Name: A localized name for the action (2) suitable for display in the client UI. This name SHOULD be relatively short for display in a list of available actions (2), in contrast to **ruleDesignerType.Sentence**, which is meant for display in situations that are less space-constrained. Unlike **WorkflowInfo.Conditions.Condition.Name**, the substring "%1" has no special meaning in this attribute.

WorkflowInfo.Actions.Action.Assembly: An implementation-specific<33> string that uniquely identifies the binary file that contains the class specified in **ClassName**.

WorkflowInfo.Actions.Action.ClassName: An implementation-specific<34> string that, in conjunction with the **Assembly** attribute, uniquely identifies the class in the binary file specified by the **Assembly** attribute that contains the implementation of this action (2).

WorkflowInfo.Actions.Action.Category: A localized name of a category that SHOULD be used by the client application to group related actions (2) in its UI. If this attribute is omitted, the client application MUST assign a default category.

WorkflowInfo.Actions.Action.CreatesTask: Indicates whether this action (2) creates a task in the list (1) of tasks for the current workflow (2). The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Description
(omitted) FALSE 0	This action (2) does not create a task in the list (1) of tasks for the running workflow (2).
TRUE 1	This action (2) creates a task in the list (1) of tasks for the running workflow (2).

WorkflowInfo.Actions.Action.CreatesInList: If present, indicates that this action (2) creates an item in a list (1) or **document library**. The value of this attribute MUST match the **parametersType.Name** attribute of the **Action.Parameters.Parameter** that specifies in which list (1) or document library the item is to be created. For example, if the **CreatesInList** attribute is set to "ToList" and there is a **Parameter** element with the **Name** attribute set to "ToList", the value of that element specifies an identifier for the list (1) in which the item is to be created. If absent, indicates that this action (2) does not create an item in a list (1) or document library.

WorkflowInfo.Actions.Action.DataSources: Each child **DataSource** element specifies a construct that a client application SHOULD make available to enable one action (2) in the UI to get data from another action (2) in the UI.

WorkflowInfo.Actions.Action.Modifications: Each child **Modification** element specifies one workflow modification that the internal workflow (2) logic MAY activate or deactivate during the run of one workflow instance.

WorkflowInfo.Actions.Action.Modifications.Modification: Specifies one workflow modification that the client SHOULD add to the internal workflow (2) logic if the user adds the action (2) to the workflow (2) through the UI.

WorkflowInfo.Actions.Action.Modifications.Modification.TypeId: Specifies the identifier that the client SHOULD add to the internal workflow (2) logic to activate and deactivate the workflow modification.

WorkflowInfo.Actions.Action.Modifications.Modification.NameFormat: Specifies the text that the client SHOULD add to the workflow (2) if this action (2) is used. It is used at runtime of the workflow (2) to display in user-accessible UI elements when beginning interaction with the workflow modification. If the client places the token {0} in the string, the internal workflow (2) logic MAY replace it with workflow (2)-logic-specific text.

WorkflowInfo.Actions.Action.Modifications.Modification.FormURN: Specifies the **Uniform Resource Name (URN)** that the client SHOULD add to the workflow (2) if this action (2) is used. It is used at runtime of the workflow (2) to specify the form to display to the user when the user is interacting with the workflow modification.

WorkflowInfo.Actions.Action.ActionVariables: An implementation-specific<35> way for an action (2) to declare some local variables that the client application SHOULD insert as child objects of the action (2) when the action (2) is inserted.

WorkflowInfo.Actions.Action.ActionBody: An implementation-specific<36> way for an action (2) to declare some actions (2) that the client application SHOULD insert as child objects of the action (2) when the action (2) is inserted.

WorkflowInfo.Actions.Action.ActionConditions: An implementation-specific<37> way for an action (2) to declare some conditional logic that the client application SHOULD insert as child objects of the action (2) when the action (2) is inserted and store in a workflow rules file.

WorkflowInfo.Actions.Action.IsError: If set to TRUE, the client application SHOULD regard every instance of this action (2) in its UI as an error.

WorkflowInfo.Actions.Action.CreatedTaskFormType: The following table describes possible values.

Value	Description
(omitted) DataCollectTask	The action (2) creates a task that collects data from one user.
GroupAssignedTask	The action (2) creates a task that collects data from one or more users.
TodoItemTask	The action (2) creates a task that does not collect data from users, but only exists for a user to validate that they have done something.
TaskProcess	The action (2) creates a task that has a form that allows for ad-hoc collaboration and might collect data from one or more users.

WorkflowInfo.Actions.Action.InvocationTagGroupRef: Specifies possible nesting of this action (2) with itself and other actions (2).

WorkflowInfo.Actions.Action.SandboxedFunction: If set to TRUE, the client application SHOULD insert an implementation-specific<38> action (2) when this action (2) is selected, and that action (2) SHOULD be configured to call a function defined by the conjunction of **AssemblyName**, **ClassName**, and **FunctionName**. If set, **AssemblyName**, **ClassName**, **FunctionName**, and **___SolutionId** MUST also be set.

WorkflowInfo.Actions.Action.FunctionName: An implementation-specific<39> string that names a function to call in the assembly and class defined by **AssemblyName** and **ClassName**.

WorkflowInfo.Actions.Action.___SolutionId: A GUID that the client application SHOULD write to the implementation-specific<40> action (2) that the server SHOULD use to help locate the assembly at runtime of the workflow (2).

WorkflowInfo.Actions.Action.AppliesTo: Specifies whether the action (2) can be used in workflows (2) associated with certain types of lists (1). For a description of possible values, see **appliesToTypes**.

WorkflowInfo.Actions.Action.ListModeration: Specifies whether the action (2) operates correctly only when used in a workflow (2) associated with a list (1) or document library that requires content approval. The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Description
(omitted) FALSE 0	This action (2) does not require a list (1) or document library that requires content approval.
TRUE 1	This action (2) operates correctly only when used in a workflow (2) associated with a list (1) or document library that requires content approval.

WorkflowInfo.Actions.Action.UsesCurrentItem: Specifies whether the action (2) requires data from an item on which the workflow (2) is currently operating, and thus can be used only in workflows (2) in which there is the concept of a current item. The client MUST ignore case when evaluating this attribute. Possible values are those specified for **WorkflowInfo.Conditions.Condition.UsesCurrentItem**.

WorkflowInfo.Actions.Action.RuleDesigner: Contains information for displaying this action (2) in the client application UI. See **ruleDesignerType**.

WorkflowInfo.Actions.Action.Parameters: Contains information about the parameters accepted by this action (2). See **parametersType**.

WorkflowInfo.CompositeSteps: Each child **CompositeStep** specifies one container that contains actions (2) that a client application SHOULD<41> recognize and for which it SHOULD provide additional functionality.

WorkflowInfo.CompositeSteps.CompositeStep: Specifies one container that contains actions (2) that a client application SHOULD<42> recognize and for which it SHOULD provide additional functionality.

WorkflowInfo.CompositeSteps.CompositeStep.DataSources: Each child **DataSource** element specifies a construct that a client application SHOULD<43> make available to enable one action (2) in the UI to get data from a composite step in the UI.

WorkflowInfo.CompositeSteps.CompositeStep.Name: Specifies the localized name that the client application SHOULD<44> display for the **CompositeStep** in the UI.

WorkflowInfo.CompositeSteps.CompositeStep.ClassName: An implementation-specific<45> string that, in conjunction with the **Assembly** attribute, uniquely identifies the class in the binary file specified by the **Assembly** attribute that contains the implementation of this composite step.

WorkflowInfo.CompositeSteps.CompositeStep.Assembly: An implementation-specific<46> string that uniquely identifies the binary file that contains the class specified in **ClassName**.

WorkflowInfo.CompositeSteps.CompositeStep.AppliesTo: Specifies whether the composite step can be used in workflows (2) associated with specific types of lists (1). For a description of possible values, see **appliesToTypes**.

WorkflowInfo.CompositeSteps.CompositeStep.Description: A string that allows a client application to show additional text describing the functioning of a **CompositeStep**.

WorkflowInfo.CompositeSteps.CompositeStep.Order: If set, the client application SHOULD<47> display adjacent **CompositeStep** elements in order sorted on this value, with the lowest numbers first.

WorkflowInfo.VariableTypes: Specifies the list of types that the client application SHOULD<48> enable as local variables in the orchestration of the workflow (2).

WorkflowInfo.VariableTypes.VariableType: Specifies one type that the client application SHOULD<49> enable as a local variable in the UI.

WorkflowInfo.VariableTypes.VariableType.Type: Specifies an implementation-specific<50> naming of the class type of the variable.

WorkflowInfo.VariableTypes.VariableType.Assembly: An implementation-specific<51>string that uniquely identifies the binary file that contains the class specified in **Type**.

WorkflowInfo.VariableTypes.VariableType.CoercionActivity: An implementation-specific<52> string that uniquely identifies a class that can convert values of any type into the type specified by **Type**.

WorkflowInfo.VariableTypes.VariableType.Default: If set to TRUE, the client application SHOULD<53> use this variable type as the default variable type in the UI. If set to FALSE, this **VariableType** element does not specify the default variable type in the UI.

WorkflowInfo.VariableTypes.VariableType.DisplayName: Specifies a localized string that the client application SHOULD<54> display in the UI for this type of variable.

WorkflowInfo.VariableTypes.VariableType.Hidden: If set to FALSE, specifies if a client application SHOULD<55> show a particular **VariableType** in the UI for creating and managing local variables. If set to TRUE, specifies if a client SHOULD NOT show a particular VariableType in the UI for creating and managing local variables.

WorkflowInfo.VariableTypes.VariableType.ItemIdType: If set to TRUE, the client application SHOULD<56> use this variable type as the type for identifying list items in lists (1). If set to FALSE, this **VariableType** element does not specify the type for identifying list items.

WorkflowInfo.AssemblyRedirects: Specifies a list of transformations from one binary file to another binary file in a versioned binary file system so that the client has sufficient information to load the desired binary file.

WorkflowInfo.AssemblyRedirects.AssemblyRedirect: Specifies one transformation from one binary file to another binary file in a versioned binary file system so that the client has sufficient information to load the desired binary file.

WorkflowInfo.AssemblyRedirects.AssemblyRedirect.From: Specifies an implementation-specific<57> binary file name. If the client receives this value, it SHOULD<58> load the binary file specified by **To** instead.

WorkflowInfo.AssemblyRedirects.AssemblyRedirect.To: Specifies an implementation-specific<59> binary file name. The client SHOULD<60> load this binary file instead of the one specified by **From**.

WorkflowInfo.Coercions: Each child element represents one capability that the client SHOULD<61> make available to convert data from one form to another, whether from one class to another or from one format within a class to another format. A simple example of this would be a coercion that does a floor operation on a floating point number.

WorkflowInfo.Coercions.Coercion: Specifies one capability of converting data from one form to another.

WorkflowInfo.Coercions.Coercion.Name: Specifies the internal name for this capability to coerce data.

WorkflowInfo.Coercions.Coercion.Assembly: An implementation-specific<62> string that specifies in which binary file on the server the capability is implemented.

WorkflowInfo.Coercions.Coercion.DisplayName: Specifies the localizable string that the client MUST display for this capability.

WorkflowInfo.Coercions.Coercion.CatchAll: If set to TRUE, the client SHOULD<63> show this capability in scenarios where either the **DestinationType** or the **DestinationFieldType** allows this capability, regardless of **SourceType** or **SourceFieldType**. If set to FALSE, the client MUST only

show this in cases where **SourceType** or **SourceFieldType** identify that this **Coercion** is to be shown, as per the descriptions of **SourceType** and **SourceFieldType**.

WorkflowInfo.Coercions.Coercion.ClassName: An implementation-specific<64> string that, in conjunction with the **Assembly** attribute, uniquely identifies the class in the binary file specified by the **Assembly** attribute that contains the implementation of this composite step.

WorkflowInfo.Coercions.Coercion.RankDefault: Specifies a value by which the client MUST sort the list of capabilities.

WorkflowInfo.Coercions.Coercion.SourceType: If set, an implementation-specific<65> string specifying a class type from which the data is converted. If set, the client MUST show this capability only when the previous value is of the matching type, unless the **SourceFieldType** or the **CatchAll** attribute specifies that it SHOULD be shown.

WorkflowInfo.Coercions.Coercion.SourceFieldType: If set, specifies the field type that the client MAY use to determine whether this capability is shown. If set, the client MUST show this capability only when the previous value is of the matching type, unless the **SourceField** or the **CatchAll** attribute specifies that it SHOULD be shown.

WorkflowInfo.Coercions.Coercion.DestinationType: An implementation-specific<66> string specifying the output class type of the value of this capability. If the destination of the value cannot handle this type, the client MUST NOT display this capability.

WorkflowInfo.Coercions.Coercion.DestinationFieldType: A field type string specifying the output field type of the value of this capability. If the destination of the value cannot handle this type, the client MUST NOT display this capability.

ruleDesignerType: The **Condition** and **Action** elements in the **WorkflowInfo** document both contain a **RuleDesigner** element that follows the same structure, described here as **ruleDesignerType**.

ruleDesignerType.Sentence: A localized string that can be used to represent the sentence in the client application workflow (2) design interface. In contrast with **WorkflowInfo.Conditions.Condition.Name** and **WorkflowInfo.Actions.Action.Name**, this string is generally longer and more descriptive and also contains placeholders for the various **FieldBind** elements. Placeholders are substrings consisting of a percent sign (%) and a positive integer, such as "%1" or "%15". The number in the placeholder corresponds with the **Id** attribute of a child **FieldBind** element. When displaying the sentence in the UI, the client application MUST replace each placeholder with a representation of the appropriate **FieldBind**. Literal percent signs can be escaped as "%%", and the client application MUST replace instances of this substring with a single literal percent sign.

For example, if the **Sentence** attribute were "Set %1 to %2", the client application would display the action (2) or condition as "Set MyProperty to 25" if the **FieldBind** element with **Id** set to 1 had a text representation of "MyProperty" and the **FieldBind** element with **Id** set to 2 had a text representation of "25".

ruleDesignerType.FieldBind: Each **FieldBind** represents a portion of the sentence for the parent **Condition** or **Action** that can be changed by the user in the client application UI. A field binding connects one portion of the UI to one or more parameters on the condition or action (2). When the value is changed in the UI, the appropriate parameters on the condition or action (2) are updated to match.

ruleDesignerType.FieldBind.Field: A comma-delimited list of parameter names. The parameters with **Name** attributes in this list are modified when the field is modified. The number of parameters

in the list depends on the **DesignerType** of the **FieldBind**. Primitive **DesignerTypes**, such as string, only modify one parameter, so the **Field** attribute would contain a single parameter name.

ruleDesignerType.FieldBind.Text: A localized string containing text to display for the **FieldBind** when there is no value set for the parameters listed in **ruleDesignerType.FieldBind.Field**. For example, if a field bind referenced an integer parameter in its **Field** attribute, and its **Text** attribute were set to "number", a client application might display that field bind as "number" before a value was set for the referenced integer parameter. After 5 was set as the value, it would instead display that field bind as "5", ignoring the value of the **Text** parameter.

ruleDesignerType.FieldBind.Id: The positive integer to be used in placeholders in the parent element **ruleDesignerType.Sentence** attribute to take the place of this **FieldBind**. For example, if the **Id** attribute of a **FieldBind** were "2", the client would display that **FieldBind** in the UI instead of the substring "%2" in the sentence for a condition or action (2).

ruleDesignerType.FieldBind.DesignerType: The client application MUST display UI elements compatible with those in the following table when editing the value of the parameters specified in the **Field** attribute of this **FieldBind**. For example, for "Date", instead of a text box as specified in the table, a compatible UI would include a calendar. The client MUST ignore case when evaluating this attribute. If the **DesignerType** attribute is omitted or is of an unknown value, the client application MUST assume "Text". Possible values are listed in the following table, along with the parameters which MUST be used and the UI element.

DesignerType	Parameters	UI element
(omitted) (unknown) Text TextBox	String	Single-line text box.
AddPermission	List of roles List of users	Dialog box that allows adding roles to users.
Assignment	Assigned to CC Comments Subject Duration Duration unit Due date	Dialog box for setting the parameters of assignment, due date, and other data.
Bool	Boolean	List containing the choices TRUE or FALSE.
CheckPermission	Single parameter	Dialog box for selecting a set of permissions.
ChooseDocLibItem	List (1) identifier List item	Document library item selection dialog box.
ChooseListItem	List (1) identifier List item	List item selection dialog box.

DesignerType	Parameters	UI element
CreateListItem	List (1) identifier Item properties Overwrite	Single-line text box with button that shows a dialog box to bind to a specific list item.
DataSourceFieldNames	Single parameter	List containing options retrieved from the ancestor **Action** or **CompositeStep** that has a **DataSource** element matching the child **DataSourceRef** element. The result of choosing an element from this list should be the literal value chosen.
DataSourceValues	Single parameter	List containing options retrieved from the ancestor **Action** or **CompositeStep** that has a **DataSource** element matching the child **DataSourceRef** element. The result of choosing an element from this list should be the runtime value of the chosen field in the datasource.
Date	Date/time	Text box that accepts dates and times.
Dependent	Single parameter	UI element shown depends upon other values in the parameter **RuleDesigner**, referencing the **TypeFrom** and **OperatorTypeFrom** attributes.
Dropdown	Single parameter	List containing options specified as **Option** child elements of the **FieldBind** element.
Email	To CC Subject Body	Dialog box for composing an e-mail.
FieldNames	Field	List of all fields in the list (1) or document library with which the workflow (2) is associated.
Float	Float	Text box accepting a floating-point number.
Hyperlink	URL	URL dialog box.
Integer	Int	Text box accepting an integer.
ListNames	List (1) identifier	List of lists (1)in the current site (2).
Operator	Value	List containing operator options specified as **Option** child elements of the **FieldBind** element. In contrast to "Dropdown", the list of operators is filtered based on the type of the operand. See **ruleDesignerType.FieldBind.OperatorTypeFrom** and **ruleDesignerType.FieldBind.Option.TypeFilter**.
ParameterNames	Name	List containing the names of variables that are defined in the current workflow (2).
Person	List of people	Dialog box for selecting accounts on the current site (2).
RemovePermission	List of roles List of users	Dialog box for removing roles from users.

DesignerType	Parameters	UI element
ReplacePermission	List of roles List of users	Dialog box for replacing the roles of users.
SinglePerson	Person	Dialog box for selecting a single account on the current site (2).
StatusDropdown	Status value	Dropdown list for entering a new status value or reusing an existing value.
StringBuilder	String	Single-line text box.
Survey	Title Content type identifier	Dialog box for building a survey form to be presented to users of the workflow (2) at runtime to collect data.
TaskSummary	Single parameter	View for setting properties of a task process.
TextArea	String	Text box.
UpdateListItem	List (1) identifier List item Item properties	Dialog box containing a list of lists (1) and fields in each list (1), enabling new values to be set for the fields in a list (1).
WritableFieldNames	List of names	List containing all fields in the list (1)or document library with which the workflow (2) is associated that are not read-only, or a list of all document libraries on the current site (2) that are not read-only, depending on the context.

For example, take the following **WorkflowInfo.Conditions.Condition.RuleDesigner** element:

```
<RuleDesigner Sentence="the file size is between %1 and %2 kilobytes">
    <FieldBind Id="1" Field="min" Text="size" DesignerType="Integer" />
    <FieldBind Id="2" Field="max" Text="size" DesignerType="Integer" />
</RuleDesigner>
```

This condition has two interactive portions when displayed in the client UI; a minimum size and a maximum size. Before any fields are set, a client application might display it as "the file size is between size and size kilobytes". Clicking the first "size" would display a textbox that accepts numeric input, because that is an appropriate UI element for a designer type of "Integer". The user types the number "1024" and commits the value. The client application sets the value of the parameter with the name "min" because the **Id** attribute of that **FieldBind** is "1" and the user clicked on the UI element that appeared in place of the "%1" in the sentence. Because there is now a value set for the first **FieldBind**, the client application would update the display to appear as "the file size is between 1024 and size kilobytes".

ruleDesignerType.FieldBind.OperatorTypeFrom: Only valid when the value of **DesignerType** is set to "Operator". If present, the value of this attribute MUST match the **Name** attribute of one of the **Parameters.Parameter** child elements of the **ruleDesignerType** parent element. The **OperatorTypeFrom** attribute specifies which parameter's type MUST be used to determine which operator options. See **ruleDesignerType.FieldBind.Option.TypeFilter**.

ruleDesignerType.FieldBind.TypeFrom: If present, the value of this attribute MUST match the **Name** attribute of one of the other **FieldBind** elements. The **TypeFrom** attribute specifies which other field binding's rule designer MUST be used for this field binding.

ruleDesignerType.FieldBind.Function: Specifies that this **FieldBind** is not a typical binding between a placeholder and one or more parameters, but that the placeholder specified by the **Id** attribute on the **FieldBind** element MUST be replaced with a server implementation-specific<67> string representation of the class implementing the action (2) or condition. This would typically be used only on a field binding for **WorkflowInfo.Actions.Default**. The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Description
(omitted) FALSE	This is a normal **FieldBind**, binding a placeholder to one or more parameters.
TRUE	The placeholder in the parent **Action** or the **Sentence** attribute of the **Condition** element that refers to this **FieldBind** MUST be replaced with a string representing the action (2) or condition.

If the **Function** attribute is set to TRUE, all other attributes on **FieldBind** are ignored, except **Id**.

For example, the following **Actions** element contains a **Default** element that uses a **FieldBind** with **Function** set to TRUE. In the following example, all unrecognized actions (2) would be displayed as "Perform unrecognized action *name of the action*".

```
<Actions>
   <Default>
      <RuleDesigner Sentence="Perform unrecognized action %1">
         <FieldBind Id="1" Function="true" />
      </RuleDesigner>
   </Default>
</Actions>
```

ruleDesignerType.FieldBind.Value: This attribute is reserved for future use and MUST be ignored by the client.

ruleDesignerType.FieldBind.Option: Each **Option** element specifies one option to be presented in a list of possible values for a **FieldBind** of certain designer types. See **ruleDesignerType.FieldBind.DesignerType**.

ruleDesignerType.FieldBind.Option.Name: A localized string to display in the client application UI that represents this option.

ruleDesignerType.FieldBind.Option.Value: When this option is selected in the client application UI, the client MUST set the value of the parameter specified in the **FieldBind.Field** attribute to the value specified in the selected **FieldBind.Option.Value** attribute.

ruleDesignerType.FieldBind.Option.TypeFilter: Only valid when the value of **DesignerType** is set to "Operator". A server implementation-specific<68> string that specifies one or more requirements for this option to be displayed in the client application UI, in conjunction with the **ruleDesignerType.FieldBind.OperatorTypeFrom** attribute. Using this value and the type of the parameter specified by **OperatorTypeFrom**, the client application MUST determine whether the option is appropriate for display. If this attribute is omitted, the client application MUST always display this option.

Using this attribute, the server can, for example, prevent operators such as "contains the substring" from appearing if the left-hand operand is not a string type.

ruleDesignerType.FieldBind.Option.UnaryHides: Valid only when the value of **DesignerType** is set to "Operator". If this attribute is set, the client application MUST hide the **FieldBind** with the name specified in the **UnaryHides** attribute when this option is selected.

For example, the following **RuleDesigner** element specifies a sentence of the form "value operator value". When the operators "is greater than" and "is less than" are chosen, all three field bindings are displayed in the UI and can have values set. However, when the operator "is null" is selected, the third field binding (the second operand) is hidden, because the "is null" option for the operator field binding has its **UnaryHides** attribute set to "right", the same as the **Field** attribute on the third field binding.

```
<RuleDesigner Sentence="%1 %2 %3">
   <FieldBind Id="1" Field="left" Text="value" />
   <FieldBind Id="2" Field="operator" DesignerType="Operator" OperatorTypeFrom="left"
Text="operator">
      <Option Name="is greater than" Value="GreaterThan" />
      <Option Name="is less than" Value="LessThan" />
      <Option Name="is null" Value="IsNull" UnaryHides="right" />
   </FieldBind>
   <FieldBind Id="3" Field="right" Text="value" />
</RuleDesigner>
```

ruleDesignerType.FieldBind.DataSourceRef: Specifies that a client application MUST get a list of possible values from **DataSource** elements that are owned by **Action**, **Condition**, or **CompositeStep** elements that are ancestors of the **Action** or **Condition** element that is the parent element of the **ruleDesignerType**. This is intended to enable dynamic population of the list of possible values, using the ability to nest an **Action** or **Condition** inside an **Action** or **CompositeStep** element, and having the ancestor provide that list of values for the descendant. If there is more than one **DataSourceRef** element as a child of a **FieldBind** element, the list of possible values is the union of the possible values.

ruleDesignerType.FieldBind.DataSourceRef.Ref: Specifies the **InternalName** of the **DataSource** from which to obtain a list of possible values.

defaultElementType: Specifies a sentence that the client application MUST use when displaying a type of workflow (2) construct that is not recognized as one of the **Condition** or **Action** elements in the **WorkflowInfo** document returned by this operation.

defaultElementType.RuleDesigner: The sentence to display in the client application representing an unrecognized workflow (2) construct. See **ruleDesignerType**.

parametersType: Several elements in the **WorkflowInfo** document contain a **Parameters** element that follows the same structure, described here as **parametersType**.

parametersType.Parameter: Each **Parameter** element specifies a single parameter of a specified **Condition** or **Action**.

parametersType.Parameter.Name: The name of this parameter referenced by **ruleDesignerType.FieldBind.Field**.

parametersType.Parameter.Type: An implementation-specific<69> string that specifies the type of this parameter.

parametersType.Parameter.Description: If set, this attribute specifies the text that the client SHOULD<70> display when this parameter is selected in a grid structure matching names of parameters with their currently set values.

parametersType.Parameter.Direction: Specifies whether the parameter is an input to or an output of the condition or action (2), and whether it is required to have a value. The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Decription
(omitted) In	This parameter is an input to the parent condition or action (2) and is required to have a value.
Out	This parameter is an output of the parent condition or action (2) and is required to have a value.
Optional	This parameter is an optional input to the parent condition or action (2) and is not required to have a value.

parametersType.Parameter.DisplayName: If set, this attribute specifies the name that the client SHOULD<71> use when showing this parameter in a grid structure matching names of parameters with their currently set values.

parametersType.Parameter.DesignerType: The client application MUST display UI elements compatible to those in the following table when editing the value of a parameter defined by the **Parameter** element. For example, for "Date", instead of a text box as specified in the table, a compatible UI would include a calendar. The client MUST ignore case when evaluating this attribute. If the **DesignerType** attribute is omitted or an unknown value, the client application MAY use additional information, including **FieldBind** child objects of the **RuleDesigner** element that is a sibling of the **parametersType** element. Otherwise, the client MUST assume "Text". The following table describes possible **DesignerType** elements, along with the parameters that they MUST be used with, and the UI elements.

DesignerType	Parameters	UI element
(omitted) *(unknown)* Text TextBox Dependent	String	Single-line text box.
ContentType	Content type identifier	Dialog box for creating or updating a content type.
DataSourceFieldNames	Single parameter	List containing options retrieved from the ancestor **Action** or **CompositeStep** that has a **DataSource** element matching the child **DataSourceRef** element. The result of choosing an element from this list should be the literal value chosen.
DataSourceValues	Single parameter	List containing options retrieved from the ancestor **Action** or **CompositeStep** that has a **DataSource** element matching the child **DataSourceRef** element. The result of choosing an element from this list should be the runtime value of the chosen field in the datasource.
Date	Date/time	Text box that accepts dates and times.

DesignerType	Parameters	UI element
Dropdown	Single parameter	List containing options specified as **Option** child elements of the **FieldBind** element.
FieldNames	Field	List of all fields in the list (1) or document library with which the workflow (2) is associated.
Hide	None	The client SHOULD NOT show this value.
Integer	Int	Text box accepting an integer.
ItemProperties	Hash table	Dialog box for setting a hash table of name-value pairs.
ListItem	List (1) identifier List item identifier	Dialog box for selecting a list item.
ListNames	List (1) identifier	List of lists (1) in the current site (2).
ParameterNames	Name	List containing the names of variables that are defined in the current workflow (2).
Person	List of people	Dialog box for selecting accounts on the current site (2).
SinglePerson	Person	Dialog box for selecting a single account on the current site (2).
StatusDropdown	Status value	Dropdown list that enables entering a new status value or reusing an existing value.
StringBuilder	String	Single-line text box.
TextArea	String	Text box.
WritableFieldNames	List of names	List containing all fields in the list (1) or document library with which the workflow (2) is associated that are not read-only, or a list of all document libraries on the current site (2) that are not read-only, depending on the context.

parametersType.Parameter.InitialBinding: If set, this attribute specifies the default variable that the client MUST use for this parameter when creating a new copy of the condition or action (2) containing this parameter. If omitted, the client MUST NOT set a default value or variable for the parameter unless **InitialValue** is set. If set, **InitalValue** MUST NOT be set.

parametersType.Parameter.InitialValue: If set, this attribute specifies the default value that the client MUST use for this parameter when creating a new copy of the condition or action (2) containing this parameter. If omitted, the client MUST NOT set a default value for the parameter unless **InitialBinding** is set. If set, **InitialBinding** MUST NOT be set.

parametersType.Parameter.ShowLookupForDropdown: If set and the DesignerType attribute indicates that the displayed UI is a drop down list, specifies that a button to lookup a value dynamically at runtime SHOULD<72> be displayed.

dataSourceType: Each element of **dataSourceType** indicates one source of data that an action (2) or composite step makes available to its descendant activities. If present, the client SHOULD

allow descendant actions (2) to find among all ancestor actions (2) and composite steps the set of data sources that they provide.<73>

A typical example is an action (2) that iterates over elements in a SQL table. One piece of data that would be of interest to the descendant activities is which item from the SQL table is the current item of iteration.

dataSourceType.SchemaSource: Each child element of the **SchemaSource** indicates how the **dataSourceType** SHOULD get the list of fields that the client SHOULD display for this data source.<74>

dataSourceType.SchemaSource.Fields: Specifies that the fields come from an embedded set of **Field** elements. Each child element specifies one field for use in the client UI for displaying data sources.

dataSourceType.SchemaSource.Fields.Field: Specifies one field for use in the client UI for displaying data sources.

dataSourceType.SchemaSource.Fields.Field.Name: Specifies the internal name of the **dataSourceType.SchemaSource.Fields.Field**.

dataSourceType.SchemaSource.Fields.Field.DisplayName: Specifies the localizable name of the **dataSourceType.SchemaSource.Fields.Field** to display to the user.

dataSourceType.SchemaSource.Fields.Field.Type: Specifies the type of data held by the **dataSourceType.SchemaSource.Fields.Field.**

dataSourceType.SchemaSource.ContentType: Specifies that the client MUST retrieve the set of fields from a content type.

dataSourceType.SchemaSource.ContentType.IDFrom: Specifies how the client MUST get the content type identifier for the content type referred to by **dataSourceType.SchemaSource.ContentType**. If the value begins with "Parent", the value is retrieved from the parent of the action (2) or composite step that declares the **dataSourceType** element, using the string that comes after "Parent". The value is then used as the name of the field from the target action (2) or composite step from which to get the data.

dataSourceType.SchemaSource.Dictionary: Specifies that the client MUST retrieve the set of fields from a dictionary of name-value pairs. The name-value pairs are delimited by semicolons, and the name and value are separated within the pair by a comma.

dataSourceType.SchemaSource.Dictionary.ValueFrom: Specifies how the client MUST get the content type identifier for the content type referred to by **dataSourceType.SchemaSource.Dictionary**. If the value begins with "Parent", the value is retrieved from the parent of the action (2) or composite step that is the ancestor of the **dataSourceType** element, using the string that comes after "Parent". The value is then used as the name of the field from the target action (2) or composite step from which to get the data.

dataSourceType.SchemaSource.NestedInitiationFieldNodes: Specifies that the client MUST retrieve the set of fields from the schema of the data used to begin the workflow (2).

dataSourceType.Evaluation: Specifies how the workflow (2) obtains the value defined by the schema at runtime of the workflow (2). The client MUST use this element and its descendants for proper authoring and round-tripping of the value.

dataSourceType.Evaluation.ActivitySource: Specifies that the value is retrieved from the fields of the action (2) or composite step that is the ancestor of the **dataSourceType** element.

dataSourceType.Evaluation.HashtableSource: Specifies that the value is retrieved from a hash table.

dataSourceType.Evaluation.HashtableSource.HashtableFrom: Specifies how the client MUST get the hash table referred to by **dataSourceType.Evaluation.HashtableSource**. If the value begins with "Parent", the value is retrieved from the parent of the action (2) or composite step that is the ancestor of the **dataSourceType** element using the string that comes after "Parent". The value is then used as the name of the field from the target action (2) or composite step from which to get the data.

dataSourceType.InternalName: Specifies the internal name for the data source, for reference from a **ruleDesignerType.FieldBind.DataSourceRef** element.

dataSourceType.Name: Specifies a localizable display name for the data source that the client MUST show to users.

dataSourceType.NameProperty: Specifies how the client MUST get an additional piece of data for differentiating the UI elements defined by the **dataSourceType** elements. If the value begins with "Parent", the value is retrieved from the parent of the action (2) or composite step that declares the **dataSourceType** element using the string that comes after "Parent". The value is then used as the name of the field from the target action (2) or composite step from which to get the data.

invocationTagAttributeGroup: Specifies declarative nesting rules for actions (2) and composite steps. Each action (2) that has either **RequiredInvocationTags** or **ProhibitedInvocationTags** set requires that an ancestor action (2) or composite step either have or not have a matching **InvocationTag** set, depending upon which attribute is used. All attributes in this attribute group are a set of semicolon-delimited values.

invocationTagAttributeGroup.RequiredInvocationTags: If set, the client MUST require that the action (2) or composite step with the attribute be a descendant of an action (2) that has an **InvocationTag** that contains this value. This value is a semicolon-delimited list of values. If more than one is set, different ancestor elements might have different **InvocationTag** attributes, as long as the union of all ancestor **InvocationTag** attributes contains all values in this list.

invocationTagAttributeGroup.InvocationTags: If set, this element provides a demarcation that descendant actions (2) or composite steps might rely on at runtime or might specifically not work with at runtime of the workflow (2). See also **ProhibitedInvocationTags** and **RequiredInvocationTags**.

invocationTagAttributeGroup.ProhibitedInvocationTags: If set, the client MUST require that the action (2) or composite step with the attribute not be a descendant of an action (2) that has an **InvocationTag** that contains this value. This value is a semicolon-delimited list of values. If more than one is set, if any ancestor actions (2) or composite steps contain that **InvocationTag**, this action (2) is not permitted to be a descendant.

appliesToTypes: Specifies whether the condition or action (2) can be used in workflows (2) associated with certain types of lists. See **WorkflowInfo.Conditions.Condition.AppliesTo** and **WorkflowInfo.Actions.Action.AppliesTo**. The client MUST ignore case when evaluating this attribute. The following table describes possible values.

Value	Description
(omitted) none	This condition or action (2) cannot be used.
list	This condition or action (2) can be used only in workflows (2) that are associated with lists

Value	Description
	(1) that are not document libraries.
doclib	This condition or action (2) can be used only in workflows (2) that are associated with document libraries.
all	This condition or action (2) can be used in any workflow (2).
displayonly	This condition or action (2) cannot be added to workflows (2), but it might validly exist in any type of workflow.

3.1.4.7.3 Complex Types

None.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 GetAssemblyMetaData

The **GetAssemblyMetaData** operation<75> retrieves a description of objects in an object-oriented system that derive from one or more base object types contained in a binary file on the server. This operation is defined as follows:

```
<wsdl:operation name="GetAssemblyMetaData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="GetAssemblyMetaDataSoapIn" />
    <wsdl:output message="GetAssemblyMetaDataSoapOut" />
</wsdl:operation>
```

The client sends a **GetAssemblyMetaDataSoapIn** request message, and the server responds with a **GetAssemblyMetaDataSoapOut** response message.

3.1.4.8.1 Messages

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

Message	Description
GetAssemblyMetaDataSoapIn	A request to initiate a **GetAssemblyMetaData** operation on the

Message	Description
	protocol server.
GetAssemblyMetaDataSoapOut	A response from the protocol server at completion of the **GetAssemblyMetaData** operation.

3.1.4.8.1.1 GetAssemblyMetaDataSoapIn

The **GetAssemblyMetaDataSoapIn** message is the request message for **GetAssemblyMetaData**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetAssemblyMetaData
```

The SOAP Body contains a **GetAssemblyMetaData** element.

3.1.4.8.1.2 GetAssemblyMetaDataSoapOut

The **GetAssemblyMetaDataSoapOut** message is the response message for **GetAssemblyMetaData**.

The SOAP Body contains a **GetAssemblyMetaDataResponse** element.

3.1.4.8.2 Elements

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

Element	Description
GetAssemblyMetaData	A request to initiate a **GetAssemblyMetaDataSoapIn** operation on the protocol server.
GetAssemblyMetaDataResponse	A response from the protocol server at completion of the **GetAssemblyMetaDataSoapOut** operation.

3.1.4.8.2.1 GetAssemblyMetaData

The **GetAssemblyMetaData** element retrieves data about which base types are of interest in which binary file. This element is defined as follows:

```
<s:element name="GetAssemblyMetaData" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="assemblyName" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="baseTypes" minOccurs="1" maxOccurs="1">
            <s:simpleType>
               <s:restriction base="s:string">
                  <s:pattern value="([^\n\r;]+;)*[^\n\r;]+ " />
               </s:restriction>
            </s:simpleType>
         </s:element>
      </s:sequence>
   </s:complexType>
```

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

assemblyName: An implementation-specific<76> string that refers to a binary file.

baseTypes: A semicolon-delimited string containing the list of base type names. This list of type names is implementation-specific<77>.

3.1.4.8.2.2 GetAssemblyMetaDataResponse

The **GetAssemblyMetaDataResponse** element is an implementation-specific description of the types that inherit from one of a set of base classes on the server in binary files. This element is defined as follows:

```
<s:element name="GetAssemblyMetaDataResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
       <s:sequence>
          <s:element name="GetAssemblyMetaDataResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
       </s:sequence>
    </s:complexType>
</s:element>
```

GetAssemblyMetaDataResult: An implementation-specific<78> return value.

3.1.4.8.3 Complex Types

None.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 GetBindingResourceData

The **GetBindingResourceData** operation<79> retrieves an implementation-specific<80> list of localized server resources for use in page editing. There might be multiple groupings of localized resources on the server, so the client MUST specify which group of resources to return using the **ResourceName** parameter. This operation is defined as follows:

```
<wsdl:operation name="GetBindingResourceData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="GetBindingResourceDataSoapIn" />
```

```
      <wsdl:output message="GetBindingResourceDataSoapOut" />
   </wsdl:operation>
```

The client sends a **GetBindingResourceDataSoapIn** request message, and the server responds with a **GetBindingResourceDataSoapOut** response message.

3.1.4.9.1 Messages

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

Message	Description
GetBindingResourceDataSoapIn	A request to initiate a **GetBindingResourceData** operation on the protocol server.
GetBindingResourceDataSoapOut	A response from the protocol server at completion of the **GetBindingResourceData** operation.

3.1.4.9.1.1 GetBindingResourceDataSoapIn

The **GetBindingResourceDataSoapIn** message is the request message for **GetBindingResourceData**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetBindingResourceData
```

The SOAP Body contains a **GetBindingResourceData** element.

3.1.4.9.1.2 GetBindingResourceDataSoapOut

The **GetBindingResourceDataSoapOut** message is the response message for **GetBindingResourceData**.

The SOAP Body contains a **GetBindingResourceDataResponse** element.

3.1.4.9.2 Elements

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

Element	Description
GetBindingResourceData	A request to initiate a **GetBindingResourceDataSoapIn** operation on the protocol server.
GetBindingResourceDataResponse	A response from the protocol server at completion of the **GetBindingResourceDataSoapOut** operation.

3.1.4.9.2.1 GetBindingResourceData

The **GetBindingResourceData** element is defined as follows:

```
<s:element name="GetBindingResourceData" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="ResourceName" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

ResourceName: The name of the group of localized resources to return.<81>

3.1.4.9.2.2   GetBindingResourceDataResponse

The **GetBindingResourceDataResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="GetBindingResourceDataResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetBindingResourceDataResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetBindingResourceDataResult: A server implementation-specific<82> list of name-value pairs of localized resources.

3.1.4.9.3   Complex Types

None.

3.1.4.9.4   Simple Types

None.

3.1.4.9.5   Attributes

None.

3.1.4.9.6   Groups

None.

3.1.4.9.7   Attribute Groups

None.

3.1.4.10   GetCustomControlList

The **GetCustomControlList** operation<83> retrieves an implementation-specific list of the controls that are available for use in pages on the server. This operation is defined as follows:

```
<wsdl:operation name="GetCustomControlList" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```

```
            <wsdl:input message="GetCustomControlListSoapIn" />
            <wsdl:output message="GetCustomControlListSoapOut" />
        </wsdl:operation>
```

The client sends a **GetCustomControlListSoapIn** request message, and the server responds with a **GetCustomControlListSoapOut** response message.

3.1.4.10.1   Messages

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

Message	Description
GetCustomControlListSoapIn	A request to initiate a **GetCustomControlList** operation on the protocol server.
GetCustomControlListSoapOut	A response from the protocol server at completion of the **GetCustomControlList** operation.

3.1.4.10.1.1   GetCustomControlListSoapIn

The **GetCustomControlListSoapIn** message is the request message for **GetCustomControlList**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetCustomControlList
```

The SOAP Body contains a **GetCustomControlList** element.

3.1.4.10.1.2   GetCustomControlListSoapOut

The **GetCustomControlListSoapOut** message is the response message for **GetCustomControlList**.

The SOAP Body contains a **GetCustomControlListResponse** element.

3.1.4.10.2   Elements

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

Element	Description
GetCustomControlList	A request to initiate a **GetCustomControlListSoapIn** operation on the protocol server.
GetCustomControlListResponse	A response from the protocol server at completion of the **GetCustomControlListSoapOut** operation.

3.1.4.10.2.1   GetCustomControlList

The **GetCustomControlList** element is defined as follows:

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

3.1.4.10.2.2 GetCustomControlListResponse

The **GetCustomControlListResponse** element MUST be returned by the server. This element is defined as follows:

```
<s:element name="GetCustomControlListResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetCustomControlListResult" minOccurs="1" maxOccurs="1">
            <s:complexType>
               <s:sequence>
                  <s:element name="ControlsList" minOccurs="1" maxOccurs="1">
                     <s:complexType>
                        <s:sequence>
                           <s:element name="Assembly" minOccurs="0" maxOccurs="unbounded">
                              <s:complexType>
                                 <s:sequence>
                                    <s:element name="Control" minOccurs="1"
maxOccurs="unbounded">
                                       <s:complexType>
                                          <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>
                                    </s:element>
                                 </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="optional"/>
                              </s:complexType>
                           </s:element>
                        </s:sequence>
                     </s:complexType>
                  </s:element>
               </s:sequence>
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetCustomControlListResult: This element MUST contain a list of controls available on the server.

ControlsList: This element MUST contain one **Assembly** element for each binary file containing controls that is installed on the server. If the technology supports organizing controls into namespaces and a binary file contains controls in more than one namespace, there MUST be one

94 / 224

Assembly element for each binary file and namespace pairing. In this case, there are multiple **Assembly** elements with the same **Name** attribute, but different **Namespace** attributes.

ControlsList.Assembly: Each instance of this element specifies one namespace in one binary file installed on the server that contains controls.<84>

ControlsList.Assembly.Name: A server implementation-specific<85> string that uniquely identifies the binary file containing controls.

ControlsList.Assembly.Namespace: The namespace in the binary file that contains the controls.

ControlsList.Assembly.TagPrefix: An implementation-specific<86> string that specifies a tag prefix to be used for controls in this binary file. When used in conjunction with other portions of implementation-specific markup, it identifies the control to use in the object-oriented system of types in this binary file. It need not be unique.

ControlsList.Assembly.Control: Each instance of this element specifies one control on the server, contained in the binary file described by the parent **Assembly** element.<87>

ControlsList.Assembly.Control.Title: An implementation-specific<88> string that serves as the most prominent identification for the control.

ControlsList.Assembly.Control.Name: This attribute MUST contain the same string as **ControlsList.Assembly.Control.Title.**

ControlsList.Assembly.Control.Description: A textual description of the control. The client MAY<89> provide this additional information to users of the control to aid the user in selecting the proper control to use from a list.

ControlsList.Assembly.Control.Template: Implementation-specific<90> markup that can be used as a template for inserting new instances of this control.

3.1.4.10.3 Complex Types

None.

3.1.4.10.4 Simple Types

None.

3.1.4.10.5 Attributes

None.

3.1.4.10.6 Groups

None.

3.1.4.10.7 Attribute Groups

None.

3.1.4.11 GetDataFromDataSourceControl

The **GetDataFromDataSourceControl** operation<91> retrieves data similar or identical to data retrieved during processing of a page. The purpose of the operation is to provide a client with information to correctly author a data-bound control. This operation is defined as follows:

```
<wsdl:operation name="GetDataFromDataSourceControl"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="GetDataFromDataSourceControlSoapIn" />
    <wsdl:output message="GetDataFromDataSourceControlSoapOut" />
</wsdl:operation>
```

The client sends a **GetDataFromDataSourceControlSoapIn** request message, and the server responds with a **GetDataFromDataSourceControlSoapOut** response message.

3.1.4.11.1 Messages

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

Message	Description
GetDataFromDataSourceControlSoapIn	A request to initiate a **GetDataFromDataSourceControl** operation on the protocol server.
GetDataFromDataSourceControlSoapOut	A response from the protocol server at completion of the **GetDataFromDataSourceControl** operation.

3.1.4.11.1.1 GetDataFromDataSourceControlSoapIn

The **GetDataFromDataSourceControlSoapIn** message is the request message for **GetDataFromDataSourceControl**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetDataFromDataSourceControl
```

The SOAP Body contains a **GetDataFromDataSourceControl** element.

3.1.4.11.1.2 GetDataFromDataSourceControlSoapOut

The **GetDataFromDataSourceControlSoapOut** message is the response message for **GetDataFromDataSourceControl**.

The SOAP Body contains a **GetDataFromDataSourceControlResponse** element.

3.1.4.11.2 Elements

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

Element	Description
GetDataFromDataSourceControl	A request to initiate a **GetDataFromDataSourceControlSoapIn** operation on the protocol server.
GetDataFromDataSourceControlResponse	A response from the protocol server at completion of the **GetDataFromDataSourceControlSoapOut** operation.

3.1.4.11.2.1 GetDataFromDataSourceControl

The **GetDataFromDataSourceControl** operation takes a string that conforms to the markup of a data source control and returns data, usable for the purposes of authoring, that matches the schema of the data that would be returned by such a control during processing on the server. It is not necessary that the data be equal or that the data be obtained in exactly the same way as it would be during processing. However, the server MUST return data of the same schema so that the client can correctly author constructs that use the data. This operation is defined as follows:

```
<s:element name="GetDataFromDataSourceControl" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="dscXml" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="contextUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

dscXml: An XML-encoded string that, after decoding, represents the full markup necessary to process the data source control. MUST include register directives if they are required by the data source control and not on the server in an implied set of register directives.

contextUrl: If non-empty, this value is a site-relative URL that the server MUST make available to the data source control to enable it to evaluate relative URL references from a base URL other than the URL of the site (2).

3.1.4.11.2.2 GetDataFromDataSourceControlResponse

The **GetDataFromDataSourceControlResponse** element is the response to **GetDataFromDataSourceControl**. It represents data that the server is returning for authoring purposes to the client. This data MUST match the schema of what would be returned at processing time. This element is defined as follows:

```
<s:element name="GetDataFromDataSourceControlResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetDataFromDataSourceControlResult" type="s:string" minOccurs="1"
maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

GetDataFromDataSourceControlResult: Either an error message or the data returned for authoring purposes. The client MUST perform an **XML decode** on the string. If the result of the XML

decode is valid XML (Extensible Markup Language), the client MUST treat this as a success case and interpret the XML as the data to use for authoring purposes. If it is not valid XML, the client MUST treat this as a failure case, and the text contained is a localized error string.

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 GetExpandedListViewXml

The **GetExpandedListViewXml** operation retrieves data similar to the data on which a view of a list (1) operates when processing a page. This operation is defined as follows:

```
<wsdl:operation name="GetExpandedListViewXml" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetExpandedListViewXmlSoapIn" />
   <wsdl:output message="GetExpandedListViewXmlSoapOut" />
</wsdl:operation>
```

The client sends a **GetExpandedListViewXmlSoapIn** request message, and the server responds with a **GetExpandedListViewXmlSoapOut** response message.<92>

3.1.4.12.1 Messages

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

Message	Description
GetExpandedListViewXmlSoapIn	A request to initiate a **GetExpandedListViewXml** operation on the protocol server.
GetExpandedListViewXmlSoapOut	A response from the protocol server at completion of the **GetExpandedListViewXml** operation.

3.1.4.12.1.1 GetExpandedListViewXmlSoapIn

The **GetExpandedListViewXmlSoapIn** message is the request message for **GetExpandedListViewXml**.

The SOAP Action value of the message is defined as follows:

http://microsoft.com/sharepoint/webpartpages/GetExpandedListViewXml

The SOAP Body contains a **GetExpandedListViewXml** element.

3.1.4.12.1.2 GetExpandedListViewXmlSoapOut

The **GetExpandedListViewXmlSoapOut** message is the response message for **GetExpandedListViewXml**.

The SOAP Body contains a **GetExpandedListViewXml** element.

3.1.4.12.2 Elements

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

Element	Description
GetExpandedListViewXml	A request to initiate a **GetExpandedListViewXmlSoapIn** operation on the protocol server.
GetExpandedListViewXmlResponse	A response from the protocol server at completion of the **GetExpandedListViewXmlSoapOut** operation.

3.1.4.12.2.1 GetExpandedListViewXml

The **GetExpandedListViewXml** operation takes a set of parameters used to define to which list (1) the view belongs, as well as additional modifiers that affect how the view appears during processing of a page. This operation is defined as follows:

```
<s:element name="GetExpandedListViewXml" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="webId" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="listViewXml" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="listGuid" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="listUrl" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="listDisplayName" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="hasConnection" type="s:boolean" />
        <s:element minOccurs="0" maxOccurs="1" name="partialView" type="s:boolean" />
      </s:sequence>
    </s:complexType>
  </s:element>
```

webId: The unique identifier of the site (2) that contains the list (1) for which to get the view. If this is empty or set to "00000000-0000-0000-0000-00000000000", the site (2) is used based on the SOAP address.

listGuid: The GUID of the list (1) for the view. If this is empty or set to "00000000-0000-0000-0000-00000000000", the **listUrl** is used instead.

listUrl: The site-relative URL, server-relative URL, or absolute URL of the root of the list (1) for the view. This value is only used if **listGuid** is empty or set to "00000000-0000-0000-0000-00000000000". If this is empty, the **listDisplayName** is used.

listDisplayName: The display name of the list (1) for the view. This is only used if **listGuid** is empty or set to "00000000-0000-0000-0000-00000000000" and **listUrl** is empty. If this is empty and used, the server returns an error.

listViewXml: An XML-encoded string. If set, after decoding it MUST conform to the specification of a **core:ViewDefinition** element, as specified in [MS-WSSCAML], section 2.3.2.17. It represents a starting point for the definition of the view of the list (1). See also **partialView**.

hasConnection: If set to TRUE, the returned **dsQueryResponse.View** element MUST contain a **FieldRef** element with a **Name** attribute set to "SelectField". If unset or set to FALSE, the server still returns **dsQueryResponse.View** containing, among other elements, a **FieldRef** element with a **Name** attribute set to "SelectField" if the view actually contains a field of that name.

partialView: If set to FALSE and **listViewXml** is set, **listViewXml** SHOULD be considered the entire view definition. If set to TRUE, the server MUST generate a full view definition and then copy the **/View/RowLimit**, **/View/ViewStyle**, and **/View/ViewFields** elements from **listViewXml** into the response, with the path **/dsQueryResponse/View**, or remove them from the response if they are absent from **listViewXml**. In any other case, the server MUST retrieve a full view definition and generate the result set from that.

3.1.4.12.2.2   GetExpandedListViewXmlResponse

The **GetExpandedListViewXmlResponse** element is defined as follows:

```
      <s:element name="GetExpandedListViewXmlResponse"
  xmlns:s="http://www.w3.org/2001/XMLSchema">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="GetExpandedListViewXmlResult"
  type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
```

GetExpandedListViewXmlResult: An XML-encoded string containing a list of the data manipulation operations that are supported. The client MUST decode it, and the data MUST conform to the following schema:

```
<s:schema elementFormDefault="qualified" xmlns:s="http://www.w3.org/2001/XMLSchema"
          xmlns:core="http://schemas.microsoft.com/sharepoint/soap/">
   <s:element name="dsQueryResponse">
      <s:complexType>
        <s:sequence>
           <s:element name="s:schema" type="s:schema" minOccurs="0" maxOccurs="1" />
           <s:element name="View" minOccurs="0" maxOccurs="1">
              <s:extension base="core:ViewDefinition">
                 <s:attribute name="CustomXsl" use="optional">
                    <s:restriction base="s:string">
                       <s:pattern value=" [^;]+(;[^;]+)+" />
                    </s:restriction>
                 </s:attribute>
```

```
                    </s:extension>
                </s:element>
                <s:element name="ParamBindings" type="s:string" minOccurs="0" maxOccurs="1" />
                <s:element name="XmlDefinition" minOccurs="0" maxOccurs="1">
                    <s:complexType>
                        <s:sequence>
                            <s:element name="View" type="core:ViewDefinition" minOccurs="0"
 maxOccurs="1" />
                        </s:sequence>
                    </s:complexType>
                </s:element>
                <s:element name="Rows">
                    <s:complexType>
                        <s:sequence minOccurs="0" maxOccurs="unbounded">
                            <s:element name="Row">
                                <s:complexType>
                                    <s:sequence />
                                    <s:anyAttribute namespace="##any" processContents="lax" />
                                </s:complexType>
                            </s:element>
                        </s:sequence>
                    </s:complexType>
                </s:element>
                <s:element name="AllowedFields">
                    <s:complexType>
                        <s:sequence>
                            <s:element name="FieldRef" minOccurs="0" maxOccurs="unbounded">
                                <s:complexType>
                                    <s:sequence />
                                    <s:attribute name="Name" type="s:string"/>
                                    <s:attribute name="Allowed" >
                                        <s:restriction base="s:integer">
                                            <s:enumeration value="0" />
                                            <s:enumeration value="1" />
                                        </s:restriction>
                                    </s:attribute>
                                </s:complexType>
                            </s:element>
                        </s:sequence>
                    </s:complexType>
                </s:element>
            </s:sequence>
            <s:attribute name="BaseViewID" type="s:integer" />
            <s:attribute name="RowLimit" type="s:integer" />
            <s:anyAttribute namespace="##any" processContents="lax" />
        </s:complexType>
    </s:element>
  </s:schema>
```

dsQueryResponse: A container element that encapsulates the information needed by a client to render the view definition.

dsQueryResponse.schema: Specifies the schema of the **dsQueryResponse.Rows** element.

dsQueryResponse.View: Specifies the view definition retrieved and sent to the client, as determined by the list (1) (see also **listGuid**, **listUrl**, and **listDisplayName**), as well as the portions of the view definition as determined by **listViewXml**, **partialView**, and **hasConnections**.

This information enables a client to render a view of the list (1) similar to that produced in the processing of a page. The server MAY<93> make additional changes to the view definition to optimize it for a client rendering operation.

dsQueryResponse.View.CustomXsl: Specifies a semi-colon delimited list of file names retrievable through a method such as **get documents** ([MS-FPSE] section 3.1.5.3.7) that contain additional XSL that a client could use to render this view.

dsQueryResponse.ParamBindings: An XML-encoded string that contains other data to match parameterized portions of the **dsQueryResponse.View** with the correct source for the data. After decoding, it MUST match the following schema:

```
<s:element name="ParameterBindings" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="ParameterBinding">
            <s:complexType>
               <s:sequence/>
               <s:attribute name="Name" type="s:string" />
               <s:attribute name="DefaultValue" type="s:string" />
               <s:anyAttribute namespace="##any" processContents="lax" />
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

dsQueryResponse.ParameterBindings: A list of **ParameterBinding** elements, each of which specifies the value for a specified parameterized portion of a view definition.

dsQueryResponse.ParameterBindings.ParameterBinding: Specifies one value available for use in a parameterized portion of a view definition.

dsQueryResponse.ParameterBindings.ParameterBinding.Name: Specifies the name of the parameterized portion to replace.

dsQueryResponse.ParameterBindings.ParameterBinding.DefaultValues: Specifies the value with which to replace the parameterized portion.

dsQueryResponse.XmlDefinition: The server MUST return this element if **partialView** is set to TRUE in the request message or if **listViewXml** is not set in the request message. The server MUST NOT return this element if **partialView** is set to FALSE in the request message and **listViewXml** is set in the request message. This represents the view definition XML previous to performing any optional expansion operations as a part of generation **dsQueryResponse.View.**

dsQueryResponse.XmlDefinition.View: Specifies the definition of the view that the server has not changed to aid in client rendering. See also **dsQueryResponse.View**.

dsQueryResponse.Rows: Specifies the data present that, when used in conjunction with the **View** element, enables a client to process the view definition and the **Rows** values into a form.

dsQueryResponse.Rows.Row: Each **Row** element specifies one data element in the dataset to use for rendering a preview of the view.

dsQueryResponse.AllowedFields: If present, indicates that potentially not all **Field** descendants of the sibling **View** element are allowed to be used. The **FieldRef** sub elements indicate whether the client should use the Field elements for rendering a view of the data.

---

102 / 224

dsQueryResponse.AllowedFields.FieldRef: Specifies one field and whether or not it is allowed to be used for rendering a view.

dsQueryResponse.AllowedFields.FieldRef.Name: The name of the field that is denoted as being allowed or not allowed.

dsQueryResponse.AllowedFields.FieldRef.Allowed: If 1, indicates the Field is allowed to be used in rendering the view. If 0, it is not allowed to be used in rendering the view.

dsQueryResponse.BaseViewID: The identifier of the **base view** for the view.

dsQueryResponse.RowLimit: Specifies the maximum number of rows of data that the definition of the view displays.

3.1.4.12.3 Complex Types

None.

3.1.4.12.4 Simple Types

None.

3.1.4.12.5 Attributes

None.

3.1.4.12.6 Groups

None.

3.1.4.12.7 Attribute Groups

None.

3.1.4.13 GetFormCapabilityFromDataSourceControl

The **GetFormCapabilityFromDataSourceControl** operation<94> retrieves a list of the data manipulation operations supported by a data source control. This operation is defined as follows:

```
<wsdl:operation name="GetFormCapabilityFromDataSourceControl"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetFormCapabilityFromDataSourceControlSoapIn" />
   <wsdl:output message="GetFormCapabilityFromDataSourceControlSoapOut" />
</wsdl:operation>
```

The client sends a **GetFormCapabilityFromDataSourceControlSoapIn** request message, and the server responds with a **GetFormCapabilityFromDataSourceControlSoapOut** response message.

3.1.4.13.1 Messages

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

Message	Description
GetFormCapabilityFromDataSourceControlSoapIn	A request to initiate a **GetFormCapabilityFromDataSourceControl** operation on the protocol server.
GetFormCapabilityFromDataSourceControlSoapOut	A response from the protocol server at completion of the **GetFormCapabilityFromDataSourceControl** operation.

3.1.4.13.1.1 GetFormCapabilityFromDataSourceControlSoapIn

The **GetFormCapabilityFromDataSourceControlSoapIn** message is the request message for **GetFormCapabilityFromDataSourceControl**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetFormCapabilityFromDataSourceControl
```

The SOAP Body contains a **GetFormCapabilityFromDataSourceControl** element.

3.1.4.13.1.2 GetFormCapabilityFromDataSourceControlSoapOut

The **GetFormCapabilityFromDataSourceControlSoapOut** message is the response message for **GetFormCapabilityFromDataSourceControl**.

The SOAP Body contains a **GetFormCapabilityFromDataSourceControlResponse** element.

3.1.4.13.2 Elements

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

Element	Description
GetFormCapabilityFromDataSourceControl	A request to initiate a **GetFormCapabilityFromDataSourceControlSoapIn** operation on the protocol server.
GetFormCapabilityFromDataSourceControlResponse	A response from the protocol server at completion of the **GetFormCapabilityFromDataSourceControlSoapOut** operation.

3.1.4.13.2.1 GetFormCapabilityFromDataSourceControl

The **GetFormCapabilityFromDataSourceControl** element takes a string that conforms to the markup of a data source control and returns the data manipulation operations that are available on the data source control. This element is defined as follows:

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

```
            <s:element name="dscXml" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

dscXml: An XML-encoded string that, after decoding, represents the full markup necessary to process the data source control. This MUST include register directives if required by the data source control. A register directive is required if the server does not have configuration that places the binary file that contains the object-oriented type in an implied set of register directives.

3.1.4.13.2.2 GetFormCapabilityFromDataSourceControlResponse

The **GetFormCapabilityFromDataSourceControlResponse** element is the response to **GetFormCapabilityFromDataSourceControl** and returns XML that specifies which types of data manipulation operations are available for the data source control. This element is defined as follows:

```
<s:element name="GetFormCapabilityFromDataSourceControlResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
        <s:sequence>
            <s:element name="GetFormCapabilityFromDataSourceControlResult" type="s:string"
minOccurs="1" maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
```

GetFormCapabilityFromDataSourceControlResult: An XML-encoded string containing a list of the supported data manipulation operations. The client MUST decode it. From the resulting XML, the root element MUST be **FormCapability**. The following table describes possible attribute values.

Name	Values	Description
CanUpdate	TRUE FALSE	If set to TRUE, the data source control enables update of existing items in the data set. If set to FALSE, the data source control does not enable update of existing items in the data set.
CanInsert	TRUE FALSE	If set to TRUE, the data source control enables addition of new items to the data set. If set to FALSE, the data source control does not enable addition of new items to the data set.
CanDelete	TRUE FALSE	If set to TRUE, the data source control enables removal of existing items from the data set. If set to FALSE, the data source control does not enable removal of existing items from the data set.

The result MUST conform to the following XSD:

```
<s:schema elementFormDefault="qualified" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:element name="FormCapability" minOccurs="1" maxOccurs="1">
        <s:complexType>
            <s:sequence>
            </s:sequence>
```

```
            <s:attribute name="CanUpdate" type="CapitalBool" use="required"/>
            <s:attribute name="CanDelete" type="CapitalBool" use="required"/>
            <s:attribute name="CanInsert" type="CapitalBool" use="required"/>
         </s:complexType>
      </s:element>
      <s:simpleType name="CapitalBool">
         <s:restriction base="s:string">
            <s:enumeration value="True"/>
            <s:enumeration value="False"/>
         </s:restriction>
      </s:simpleType>
   </s:schema>
```

3.1.4.13.3 Complex Types

None.

3.1.4.13.4 Simple Types

None.

3.1.4.13.5 Attributes

None.

3.1.4.13.6 Groups

None.

3.1.4.13.7 Attribute Groups

None.

3.1.4.14 GetSafeAssemblyInfo

The **GetSafeAssemblyInfo** operation<95> retrieves the full names of a set of assemblies on the server. The manner in which the set is specified is implementation-specific.<96> This operation is defined as follows:

```
<wsdl:operation name="GetSafeAssemblyInfo" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetSafeAssemblyInfoSoapIn"/>
   <wsdl:output message="GetSafeAssemblyInfoSoapOut"/>
</wsdl:operation>
```

The client sends a **GetSafeAssemblyInfoSoapIn** request message, and the server responds with a **GetSafeAssemblyInfoSoapOut** response message.

3.1.4.14.1 Messages

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

Message	Description
GetSafeAssemblyInfoSoapIn	A request to initiate a **GetSafeAssemblyInfo** operation on the protocol server.
GetSafeAssemblyInfoSoapOut	A response from the protocol server at completion of the **GetSafeAssemblyInfo** operation.

3.1.4.14.1.1 GetSafeAssemblyInfoSoapIn

The **GetSafeAssemblyInfoSoapIn** message is the request message for **GetSafeAssemblyInfo**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetSafeAssemblyInfo
```

The SOAP Body contains a **GetSafeAssemblyInfo** element.

3.1.4.14.1.2 GetSafeAssemblyInfoSoapOut

The **GetSafeAssemblyInfoSoapOut** message is the response message for **GetSafeAssemblyInfo**.

The SOAP Body contains a **GetSafeAssemblyInfoResponse** element.

3.1.4.14.2 Elements

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

Element	Description
GetSafeAssemblyInfo	A request to initiate a **GetSafeAssemblyInfoSoapIn** operation on the protocol server.
GetSafeAssemblyInfoResponse	A response from the protocol server at completion of the **GetSafeAssemblyInfoSoapOut** operation.

3.1.4.14.2.1 GetSafeAssemblyInfo

The **GetSafeAssemblyInfo** element requests the list of assemblies. This element is defined as follows:

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

3.1.4.14.2.2 GetSafeAssemblyInfoResponse

The **GetSafeAssemblyInfoResponse** element MUST be sent by the server when the client sends the **GetSafeAssemblyInfo** message. This element is defined as follows:

```
<s:element name="GetSafeAssemblyInfoResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
```

```
        <s:complexType>
            <s:sequence>
                <s:element name="GetSafeAssemblyInfoResult" minOccurs="1" maxOccurs="1">
                    <s:complexType>
                        <s:sequence>
                            <s:element name="SafeAssemblyInfo" minOccurs="1" maxOccurs="1">
                                <s:complexType mixed="true">
                                    <s:sequence>
                                        <s:element name="Assembly" minOccurs="0" maxOccurs="unbounded" >
                                            <s:complexType>
                                                <s:sequence />
                                                <s:attribute name="Name" type="s:string" use="required"/>
                                                <s:attribute name="FullName" type="s:string"
  use="required"/>
                                            </s:complexType>
                                        </s:element>
                                    </s:sequence>
                                </s:complexType>
                            </s:element>
                        </s:sequence>
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
```

GetSafeAssemblyInfoResult: The sub-elements of **GetSafeAssemblyInfoResult** have the following meaning:

SafeAssemblyInfo: The container for the **Assembly** elements.

SafeAssemblyInfo.Assembly: Specifies the relationship between a short name of a binary file and the full name of the binary file.

SafeAssemblyInfo.Assembly.Name: Specifies a short alias for the binary file.

SafeAssemblyInfo.Assembly.FullName: Specifies the complete name of the binary file.

3.1.4.14.3   Complex Types

None.

3.1.4.14.4   Simple Types

None.

3.1.4.14.5   Attributes

None.

3.1.4.14.6   Groups

None.

3.1.4.14.7 Attribute Groups

None.

3.1.4.15 GetWebPart

The **GetWebPart** operation retrieves the markup of a Web Part from any Web Part zone in a view of a page. This operation is defined as follows:

```
<wsdl:operation name="GetWebPart" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartSoapIn" />
   <wsdl:output message="GetWebPartSoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartSoapIn** request message, and the server responds with a **GetWebPartSoapOut** response message.

3.1.4.15.1 Messages

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

Message	Description
GetWebPartSoapIn	A request to initiate a **GetWebPart** operation on the protocol server.
GetWebPartSoapOut	A response from the protocol server at completion of the **GetWebPart** operation.

3.1.4.15.1.1 GetWebPartSoapIn

The **GetWebPartSoapIn** message is the request message for **GetWebPart**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPart
```

The SOAP Body contains a **GetWebPart** element.

3.1.4.15.1.2 GetWebPartSoapOut

The **GetWebPartSoapOut** message is the response message for **GetWebPart**.

The SOAP Body contains a **GetWebPartResponse** element.

3.1.4.15.2 Elements

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

Element	Description
GetWebPart	A request to initiate a **GetWebPartSoapIn** operation on the protocol server.

Element	Description
GetWebPartResponse	A response from the protocol server at completion of the **GetWebPartSoapOut** operation.

3.1.4.15.2.1 GetWebPart

The **GetWebPart** element requests the markup of a Web Part in a view of a page. The Web Parts MUST be serialized in the DWP format. This element is defined as follows:

```
<s:element name="GetWebPart" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageurl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

pageurl: The location of the page. MUST be formatted as an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection.

storageKey: A GUID that uniquely identifies the Web Part and specifies the Web Part for which to return the markup. MUST correspond to a Web Part that is specified in one of the Web Part zones in the view specified by the **storage** parameter. The Web Part MUST also be serializable in the DWP format.

storage: Specifies whether the Web Part is retrieved from the public view or the current user's personal view. If not set, the server MUST interpret it as "Personal".

3.1.4.15.2.2 GetWebPartResponse

The **GetWebPartResponse** element MUST be sent by the server in response to **GetWebPart**. This element is defined as follows:

```
<s:element name="GetWebPartResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartResult" type="s:string" minOccurs="1" maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartResult: An XML-encoded string, that after decoding, is in DWP format, and is the markup of the Web Part requested.

3.1.4.15.3 Complex Types

None.

3.1.4.15.4 Simple Types

None.

3.1.4.15.5 Attributes

None.

3.1.4.15.6 Groups

None.

3.1.4.15.7 Attribute Groups

None.

3.1.4.16 GetWebPart2

The **GetWebPart2** operation<97> retrieves the markup of a Web Part from any Web Part zone in a view of a page. This operation supports a greater range of returned formats than the **GetWebPart** operation and provides the behavior parameter for compatibility purposes. This operation is defined as follows:

```
<wsdl:operation name="GetWebPart2" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPart2SoapIn" />
   <wsdl:output message="GetWebPart2SoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPart2SoapIn** request message, and the server responds with a **GetWebPart2SoapOut** response message.

3.1.4.16.1 Messages

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

Message	Description
GetWebPart2SoapIn	A request to initiate a **GetWebPart2** operation on the protocol server.
GetWebPart2SoapOut	A response from the protocol server at completion of the **GetWebPart2** operation.

3.1.4.16.1.1 GetWebPart2SoapIn

The **GetWebPart2SoapIn** message is the request message for **GetWebPart2**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPart2
```

The SOAP Body contains a **GetWebPart2** element.

3.1.4.16.1.2 GetWebPart2SoapOut

The **GetWebPart2SoapOut** message is the response message for **GetWebPart2**.

The SOAP Body contains a **GetWebPart2Response** element.

3.1.4.16.2 Elements

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

Element	Description
GetWebPart2	A request to initiate a **GetWebPart2SoapIn** operation on the protocol server.
GetWebPart2Response	A response from the protocol server at completion of the **GetWebPart2SoapOut** operation.

3.1.4.16.2.1 GetWebPart2

The **GetWebPart2** element requests the markup of a Web Part in a Web Part zone in a view of a page. This element is defined as follows:

```
<s:element name="GetWebPart2" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageurl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
         <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

pageurl: The location of the page. This MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection.

storageKey: A GUID that uniquely identifies the Web Part and specifies the Web Part for which to return the markup. This MUST correspond to a Web Part that is specified in one of the Web Part zones in the view specified by the **storage** parameter. The Web Part MUST also be serializable in one of the formats supported by this operation.

storage: Specifies whether the Web Part is retrieved from the public view or the current user's personal view of the page. If not set, the server MUST interpret it as "Personal".

behavior: MUST be set to "Version3" or omitted. Explicitly setting this value indicates that in future versions of the protocol, the caller expects the behavior specified in this version of the protocol. Omitting the value indicates the server MUST have the behavior of the most recently released version of the protocol.

3.1.4.16.2.2 GetWebPart2Response

The **GetWebPart2Response** element MUST be sent by the server MUST in response to
GetWebPart2. This element is defined as follows:

```
<s:element name="GetWebPart2Response" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPart2Result" type="s:string" minOccurs="1" maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPart2Result: An XML-encoded string, that after decoding, is in one of the formats<98>
supported by this operation. It is the markup of the Web Part requested. If the Web Part is not
serializable in one of the formats supported by this operation, the server returns an error.

3.1.4.16.3 Complex Types

None.

3.1.4.16.4 Simple Types

None.

3.1.4.16.5 Attributes

None.

3.1.4.16.6 Groups

None.

3.1.4.16.7 Attribute Groups

None.

3.1.4.17 GetWebPartCrossPageCompatibility

The **GetWebPartCrossPageCompatibility** operation returns to the client data about Web Part
connections that can be authored. In particular, this operation returns data related to the authoring
of cross-page Web Part connections. The server returns the set of possible Web Part connections
that could potentially be authored between the specified Web Part from the page specified by the
sourcePageUrl or a **sourcePageContents** parameter and any of the Web Parts available on the
page specified by the **targetPageUrl** or the **targetPageContents** parameter. The client then has
the information to change an in-memory representation of the pages to author such a Web Part
connection. This element is defined as follows:

```
<wsdl:operation name="GetWebPartCrossPageCompatibility"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartCrossPageCompatibilitySoapIn" />
   <wsdl:output message="GetWebPartCrossPageCompatibilitySoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartCrossPageCompatibilitySoapIn** request message, and the server responds with a **GetWebPartCrossPageCompatibilitySoapOut** response message.

3.1.4.17.1 Messages

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

Message	Description
GetWebPartCrossPageCompatibilitySoapIn	A request to initiate a **GetWebPartCrossPageCompatibility** operation on the protocol server.
GetWebPartCrossPageCompatibilitySoapOut	A response from the protocol server at completion of the **GetWebPartCrossPageCompatibility** operation.

3.1.4.17.1.1 GetWebPartCrossPageCompatibilitySoapIn

The **GetWebPartCrossPageCompatibilitySoapIn** message is the request message for **GetWebPartCrossPageCompatibility**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartCrossPageCompatibility
```

The SOAP Body contains a **GetWebPartCrossPageCompatibility** element.

3.1.4.17.1.2 GetWebPartCrossPageCompatibilitySoapOut

The **GetWebPartCrossPageCompatibiilitySoapOut** message is the response message for **GetWebPartCrossPageCompatibility**.

The SOAP Body contains a **GetWebPartCrossPageCompatibilityResponse** element.

3.1.4.17.2 Elements

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

Element	Description
GetWebPartCrossPageCompatibility	A request to initiate a **GetWebPartCrossPageCompatibilitySoapIn** operation on the protocol server.
GetWebPartCrossPageCompatibilityResponse	A response from the protocol server at completion of the **GetWebPartCrossPageCompatibilitySoapOut** operation.

3.1.4.17.2.1 GetWebPartCrossPageCompatibility

The **GetWebPartCrossPageCompatibility** element is defined as follows:

```
<s:element name="GetWebPartCrossPageCompatibility"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="sourcePageUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
         <s:element name="sourcePageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
         <s:element name="targetPageUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
         <s:element name="targetPageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
         <s:element name="providerPartID" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="lcid" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

sourcePageUrl: The URL of the source page about which to retrieve information. If this is non-null and non-empty, **sourcePageContents** MUST be either null or empty. The location specified MUST be in the current site (2). This is used when querying the version of a page on the server for connection information.

sourcePageContents: The contents of the source page about which to retrieve information. If this is non-null and non-empty, **sourcePageUrl** MUST be either null or empty. This is used when querying the version of the page on the client for connection information.

targetPageUrl: The URL of the target page about which to retrieve information. If this is non-null and non-empty, **targetPageContents** MUST be either null or empty. The location specified MUST be in the current site (2). This is used when querying the version of a page on the server for connection information.

targetPageContents: The contents of the target page about which to retrieve information. If this is non-null and non-empty, **targetPageUrl** MUST be either null or empty. This is used when querying the version of the page on the client for connection information.

providerPartID: The identifier of the Web Part on the page specified by either **sourcePageUrl** or **sourcePageContents** to use as one of the Web Parts present in all the Web Part connection interfaces returned by this operation.

lcid: The **language code identifier (LCID)** for the language to use for all localized strings in the connection information.

3.1.4.17.2.2 GetWebPartCrossPageCompatibilityResponse

The **GetWebPartCrossPageCompatibilityResponse** element is the response sent by the server when the client calls **GetWebPartPageCrossPageCompatibility**. This element is defined as follows:

```
<s:element name="GetWebPartCrossPageCompatibilityResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartCrossPageCompatibilityResult" type="s:string"
minOccurs="1" maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartCrossPageCompatibilityResult: An XML-encoded string. After decoding, the XML MUST match the Web Part Connection Information Schema as defined in section 2.2.3.2.

3.1.4.17.3 Complex Types

None.

3.1.4.17.4 Simple Types

None.

3.1.4.17.5 Attributes

None.

3.1.4.17.6 Groups

None.

3.1.4.17.7 Attribute Groups

None.

3.1.4.18 GetWebPartPage

The **GetWebPartPage** operation<99> retrieves the implementation-specific<100> markup for a page, as well as the metadata of the page. The server MAY include additional annotations inline in the markup returned. This operation also accepts an additional parameter to request behavior for a specified protocol version. This operation is defined as follows:

```
<wsdl:operation name="GetWebPartPage" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="GetWebPartPageSoapIn" />
    <wsdl:output message="GetWebPartPageSoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartPageSoapIn** request message, and the server responds with a **GetWebPartPageSoapOut** response message.

3.1.4.18.1 Messages

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

Message	Description
GetWebPartPageSoapIn	A request to initiate a **GetWebPartPage** operation on the protocol server.
GetWebPartPageSoapOut	A response from the protocol server at completion of the **GetWebPartPage** operation.

3.1.4.18.1.1 GetWebPartPageSoapIn

The **GetWebPartPageSoapIn** message is the request message for **GetWebPartPage**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartPage
```

The SOAP Body contains a **GetWebPartPage** element.

3.1.4.18.1.2 GetWebPartPageSoapOut

The **GetWebPartPageSoapOut** message is the response message for **GetWebPartPage**.

The SOAP Body contains a **GetWebPartPageResponse** element.

3.1.4.18.2 Elements

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

Element	Description
GetWebPartPage	A request to initiate a **GetWebPartPageSoapIn** operation on the protocol server.
GetWebPartPageResponse	A response from the protocol server at completion of the **GetWebPartPageSoapOut** operation.

3.1.4.18.2.1 GetWebPartPage

The **GetWebPartPage** element is defined as follows:

```
<s:element name="GetWebPartPage" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="documentName" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

documentName: The location of the page to download. MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection. It MUST NOT begin with two periods (..).

behavior: MUST be set to "Version3" or omitted. Explicitly setting this value indicates that in future versions of the protocol, the caller expects the behavior specified in this version of the protocol.

3.1.4.18.2.2 GetWebPartPageResponse

The **GetWebPartPageResponse** element MUST be returned by the server. This element is defined as follows:

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

```
      <s:sequence>
         <s:element name="GetWebPartPageResult" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartPageResult: An XML-encoded string containing text in the same format as the **document** return value of a Server Response in HTML Mode to the **get document** method, as specified in [MS-FPSE], sections 2.1.2 and 3.1.5.3.6, respectively. In this way, the client can use similar or identical logic for processing both the value returned by this operation and the value returned by the **get document** FPSE method.

The server MAY<101> include additional implementation-specific annotations in the page markup returned by this operation that are not in markup returned by the **get document** FPSE method.

3.1.4.18.3   Complex Types

None.

3.1.4.18.4   Simple Types

None.

3.1.4.18.5   Attributes

None.

3.1.4.18.6   Groups

None.

3.1.4.18.7   Attribute Groups

None.

3.1.4.19   GetWebPartPageConnectionInfo

The **GetWebPartPageConnectionInfo** operation obtains information about the possible Web Part connections on a single page for the client authoring of those Web Part connections. The data returned to the client about this transaction can be used by the client to change the contents of the page to add or modify Web Part connections. At the time the page is processed, the Web Part connections run in the context of either the server that is returning the page or the client that is receiving the page. This operation is defined as follows:

```
<wsdl:operation name="GetWebPartPageConnectionInfo"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartPageConnectionInfoSoapIn" />
   <wsdl:output message="GetWebPartPageConnectionInfoSoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartPageConnectionInfoSoapIn** request message, and the server responds with a **GetWebPartPageConnectionInfoSoapOut** response message, as described in the following sections.

3.1.4.19.1 Messages

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

Message	Description
GetWebPartPageConnectionInfoSoapIn	A request to initiate a **GetWebPartPageConnectionInfo** operation on the protocol server.
GetWebPartPageConnectionInfoSoapOut	A response from the protocol server at completion of the **GetWebPartPageConnectionInfo** operation.

3.1.4.19.1.1 GetWebPartPageConnectionInfoSoapIn

The **GetWebPartPageConnectionInfoSoapIn** message is the request message for **GetWebPartPageConnectionInfo**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartPageConnectionInfo
```

The SOAP Body contains a **GetWebPartPageConnectionInfo** element.

3.1.4.19.1.2 GetWebPartPageConnectionInfoSoapOut

The **GetWebPartPageConnectionInfoSoapOut** message is the response message for **GetWebPartPageConnectionInfo**.

The SOAP Body contains a **GetWebPartPageConnectionInfoResponse** element.

3.1.4.19.2 Elements

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

Element	Description
GetWebPartPageConnectionInfo	A request to initiate a **GetWebPartPageConnectionInfoSoapIn** operation on the protocol server.
GetWebPartPageConnectionInfoResponse	A response from the protocol server at completion of the **GetWebPartPageConnectionInfoSoapOut** operation.

3.1.4.19.2.1 GetWebPartPageConnectionInfo

The **GetWebPartPageConnectionInfo** element obtains information necessary to author Web Part connections. This element is defined as follows:

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

119 / 224

```
              <s:element name="sourcePageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
              <s:element name="providerPartID" type="s:string" minOccurs="1" maxOccurs="1"/>
              <s:element name="lcid" type="s:string" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
      </s:complexType>
  </s:element>
```

sourcePageUrl: The URL of the page about which to retrieve information. If this is non-null and non-empty, **sourcePageContents** MUST be either null or empty. The location specified MUST be in the current site (2). This is used when querying the version of a page that already exists on the server for connection information.

sourcePageContents: The contents of the page about which to retrieve information. If this is non-null and non-empty, **sourcePageUrl** MUST be either null or empty. This is used when querying the version of the page on the client for connection information.

providerPartID: The identifier of the Web Part on the page specified by either **sourcePageUrl** or **sourcePageContents** to use as one of the Web Parts present in all the Web Part connection interfaces returned by this operation.

lcid: The identifier for the language to use for all localized strings in the returned data.

3.1.4.19.2.2 GetWebPartPageConnectionInfoResponse

The **GetWebPartPageConnectionInfoResponse** element is the response to a **GetWebPartPageConnectionInfo** request. It contains the information about available Web Part connection types to be used for creating Web Part connections. This element is defined as follows:

```
<s:element name="GetWebPartPageConnectionInfoResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
       <s:sequence>
          <s:element name="GetWebPartPageConnectionInfoResult" type="s:string" minOccurs="1"
maxOccurs="1" />
       </s:sequence>
    </s:complexType>
</s:element>
```

GetWebPartPageConnectionInfoResult: An XML-encoded string, that, after decoding, matches the Web Part Connection Information Schema, as defined in section 2.2.3.2.

3.1.4.19.3 Complex Types

None.

3.1.4.19.4 Simple Types

None.

3.1.4.19.5 Attributes

None.

3.1.4.19.6 Groups

None.

3.1.4.19.7 Attribute Groups

None.

3.1.4.20 GetWebPartPageDocument

The **GetWebPartPageDocument** operation retrieves the implementation-specific markup for a page, as well as the metadata of the page. The server MAY include additional annotations inline in the markup returned. The annotated markup returned from this operation MAY<102> differ from the annotated markup returned by the **GetWebPartPage** operation.

This operation is provided for compatibility with older versions of this protocol and is obsolete in this version of the protocol. Implementers of new clients SHOULD call **GetWebPartPage** instead. The server SHOULD<103> disallow this operation in cases in which it is not necessary for backward compatibility.

This operation is defined as follows:

```
<wsdl:operation name="GetWebPartPageDocument" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartPageDocumentSoapIn" />
   <wsdl:output message="GetWebPartPageDocumentSoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartPageDocumentSoapIn** request message, and the server responds with a **GetWebPartPageDocumentSoapOut** response message.

3.1.4.20.1 Messages

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

Message	Description
GetWebPartPageDocumentSoapIn	A request to initiate a **GetWebPartPageDocument** operation on the protocol server.
GetWebPartPageDocumentSoapOut	A response from the protocol server at completion of the **GetWebPartPageDocument** operation.

3.1.4.20.1.1 GetWebPartPageDocumentSoapIn

The **GetWebPartPageDocumentSoapIn** message is the request message for **GetWebPartPageDocument**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartPageDocument
```

The SOAP Body contains a **GetWebPartPageDocument** element.

3.1.4.20.1.2 GetWebPartPageDocumentSoapOut

The **GetWebPartPageDocumentSoapOut** message is the response message for **GetWebPartPageDocument**.

The SOAP Body contains a **GetWebPartPageDocumentResponse** element.

3.1.4.20.2 Elements

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

Element	Description
GetWebPartPageDocument	A request to initiate a **GetWebPartPageDocumentSoapIn** operation on the protocol server.
GetWebPartPageDocumentResponse	A response from the protocol server at completion of the **GetWebPartPageDocumentSoapOut** operation.

3.1.4.20.2.1 GetWebPartPageDocument

The **GetWebPartPageDocument** element is defined as follows:

```
<s:element name="GetWebPartPageDocument" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="documentName" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

documentName: The location of the page to download. MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection. It MUST NOT begin with two periods (..).

3.1.4.20.2.2 GetWebPartPageDocumentResponse

The **GetWebPartPageResponse** element MUST be returned by the server in response to **GetWebPartPageDocument**. This element is defined as follows:

```
<s:element name="GetWebPartPageDocumentResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartPageDocumentResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartPageDocumentResult: The return value for this operation is of the same format as the return value of **GetWebPartPageResult**. If implementation-specific annotations are made to the document retrieved by this operation, the server MAY<104> return different results for

GetWebPartPageDocument from those returned for **GetWebPartPage**, or it can return the same results.

3.1.4.20.3 Complex Types

None.

3.1.4.20.4 Simple Types

None.

3.1.4.20.5 Attributes

None.

3.1.4.20.6 Groups

None.

3.1.4.20.7 Attribute Groups

None.

3.1.4.21 GetWebPartProperties

The **GetWebPartProperties** operation retrieves the markup for all Web Parts from a page. It returns all Web Parts in Web Part zones for a view of the page, as well as all Web Parts outside Web Part zones. This operation is defined as follows:

```
<wsdl:operation name="GetWebPartProperties" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartPropertiesSoapIn" />
   <wsdl:output message="GetWebPartPropertiesSoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartPropertiesSoapIn** request message, and the server responds with a **GetWebPartPropertiesSoapOut** response message.

3.1.4.21.1 Messages

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

Message	Description
GetWebPartPropertiesSoapIn	A request to initiate a **GetWebPartProperties** operation on the protocol server.
GetWebPartPropertiesSoapOut	A response from the protocol server at completion of the **GetWebPartProperties** operation.

3.1.4.21.1.1 GetWebPartPropertiesSoapIn

The **GetWebPartPageDocumentSoapIn** The **GetWebPartPropertiesSoapIn** message is the request message for **GetWebPartProperties**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties
```

The SOAP Body contains a **GetWebPartProperties** element.

3.1.4.21.1.2 GetWebPartPropertiesSoapOut

The **GetWebPartPropertiesSoapOut** message is the response message for **GetWebPartProperties**.

The SOAP Body contains a **GetWebPartPropertiesResponse** element.

3.1.4.21.2 Elements

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

Element	Description
GetWebPartProperties	A request to initiate a **GetWebPartPropertiesSoapIn** operation on the protocol server.
GetWebPartPropertiesResponse	A response from the protocol server at completion of the **GetWebPartPropertiesSoapOut** operation.

3.1.4.21.2.1 GetWebPartProperties

The **GetWebPartProperties** element retrieves all Web Parts in the page. This includes all Web Parts in Web Part zones for a view of the page, as well as all Web Parts outside Web Part zones. The Web Parts MUST be serialized in the DWP format. This element is defined as follows:

```
<s:element name="GetWebPartProperties" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

pageUrl: The location of the page. It MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2).

storage: Specifies whether to retrieve the set of Web Parts to return from the public view or the current user's personal view of the page. If not set, the server MUST interpret it as "Personal".

3.1.4.21.2.2 GetWebPartPropertiesResponse

The **GetWebPartPropertiesResponse** element MUST be sent by the server in response to a **GetWebPartProperties** request. It includes all Web Parts in the requested view of the page, as well as all Web Parts in the page outside Web Part zones. This element is defined as follows:

```
<s:element name="GetWebPartPropertiesResponse" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartPropertiesResult" minOccurs="1" maxOccurs="1">
            <s:complexType mixed="true">
               <s:sequence>
                  <s:element name="WebParts" minOccurs="1" maxOccurs="1">
                     <s:complexType mixed="true">
                        <s:sequence>
                           <s:element name="WebPart" type="wpv2:WebPart" minOccurs="0"
maxOccurs="unbounded" />
                        </s:sequence>
                     </s:complexType>
                  </s:element>
               </s:sequence>
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartPropertiesResult: Specifies the outer container of the result set of this operation.

GetWebPartPropertiesResult.WebParts: Each child node of this element is a serialized Web Part in the DWP format. XML **comments** in the **WebParts** element of the matching the format <!-- .* --> indicate errors, and the content of the comment is the localized error string.

GetWebPartPropertiesResult.WebParts.WebPart: This element is a serialized Web Part in the DWP format.

3.1.4.21.3 Complex Types

None.

3.1.4.21.4 Simple Types

None.

3.1.4.21.5 Attributes

None.

3.1.4.21.6 Groups

None.

3.1.4.21.7 Attribute Groups

None.

3.1.4.22 GetWebPartProperties2

The **GetWebPartProperties2** operation<105> retrieves the markup for all Web Parts in a view of a page. This operation enables a greater range of returned formats than the **GetWebPartProperties**

operation and provides the behavior parameter for compatibility. This operation is defined as follows:

```
<wsdl:operation name="GetWebPartProperties2" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="GetWebPartProperties2SoapIn" />
   <wsdl:output message="GetWebPartProperties2SoapOut" />
</wsdl:operation>
```

The client sends a **GetWebPartProperties2SoapIn** request message, and the server responds with a **GetWebPartProperties2SoapOut** response message.

3.1.4.22.1 Messages

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

Message	Description
GetWebPartProperties2SoapIn	A request to initiate a **GetWebPartProperties2** operation on the protocol server.
GetWebPartProperties2SoapOut	A response from the protocol server at completion of the **GetWebPartProperties2** operation.

3.1.4.22.1.1 GetWebPartProperties2SoapIn

The **GetWebPartProperties2SoapIn** message is the request message for **GetWebPartProperties2**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties2
```

The SOAP Body contains a **GetWebPartProperties2** element.

3.1.4.22.1.2 GetWebPartProperties2SoapOut

The **GetWebPartProperties2SoapOut** message is the response message for **GetWebPartProperties2**.

The SOAP Body contains a **GetWebPartProperties2Response** element.

3.1.4.22.2 Elements

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

Element	Description
GetWebPartProperties2	A request to initiate a **GetWebPartProperties2SoapIn** operation on the protocol server.
GetWebPartProperties2Response	A response from the protocol server at completion of the **GetWebPartProperties2SoapOut** operation.

3.1.4.22.2.1 GetWebPartProperties2

The **GetWebPartProperties2** element retrieves all Web Parts on the page. This includes all Web Parts in Web Part zones for a view of the page, as well as all Web Parts outside Web Part zones. This element is defined as follows:

```
<s:element name="GetWebPartProperties2" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
         <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

pageUrl: The location of the page. It MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2).

storage: Specifies whether to retrieve the set of Web Parts to return from the public view or the current user's personal view of the page. If not set, the server MUST interpret it as "Personal".

behavior: MUST be set to "Version3" or omitted. Explicitly setting this value indicates that in future versions of the protocol, the caller expects the behavior specified in this version of the protocol.

3.1.4.22.2.2 GetWebPartProperties2Response

The **GetWebPartProperties2Response** element MUST be sent by the server in response to a request for all Web Parts in the requested view of the page. This includes all Web Parts in Web Part zones for a view of the page, as well as all Web Parts outside Web Part zones. This element is defined as follows:

```
<s:element name="GetWebPartProperties2Response" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartProperties2Result" minOccurs="1" maxOccurs="1">
            <s:complexType mixed="true">
               <s:sequence>
                  <s:element name="WebParts" minOccurs="1" maxOccurs="1">
                     <s:complexType mixed="true">
                        <s:sequence>
                           <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
                        </s:sequence>
                     </s:complexType>
                  </s:element>
               </s:sequence>
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

GetWebPartProperties2Result: Specifies the outer container of the result set of this operation.

GetWebPartProperties2Result.WebParts: Each child node of this element is a serialized Web Part in a supported format.<106> XML comments in the **WebParts** element indicate errors, and the content of the comment is the localized error string.

3.1.4.22.3 Complex Types

None.

3.1.4.22.4 Simple Types

None.

3.1.4.22.5 Attributes

None.

3.1.4.22.6 Groups

None.

3.1.4.22.7 Attribute Groups

None.

3.1.4.23 GetXmlDataFromDataSource

The **GetXmlDataFromDataSource** operation retrieves data from a data store for which the query is in pure XML format. This operation is defined as follows:

```
<wsdl:operation name="GetXmlDataFromDataSource"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="GetXmlDataFromDataSourceSoapIn" />
    <wsdl:output message="GetXmlDataFromDataSourceSoapOut" />
</wsdl:operation>
```

The client sends a **GetXmlDataFromDataSourceSoapIn** request message, and the server responds with a **GetXmlDataFromDataSourceSoapOut** response message.

3.1.4.23.1 Messages

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

Message	Description
GetXmlDataFromDataSourceSoapIn	A request to initiate a **GetXmlDataFromDataSource** operation on the protocol server.
GetXmlDataFromDataSourceSoapOut	A response from the protocol server at completion of the **GetXmlDataFromDataSource** operation.

3.1.4.23.1.1 GetXmlDataFromDataSourceSoapIn

The **GetXmlDataFromDataSourceSoapIn** message is the request message for **GetXmlDataFromDataSource**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/GetXmlDataFromDataSource
```

The SOAP Body contains a **GetXmlDataFromDataSource** element.

3.1.4.23.1.2 GetXmlDataFromDataSourceSoapOut

The **GetXmlDataFromDataSourceSoapOut** message is the response message for **GetXmlDataFromDataSource**.

The SOAP Body contains a **GetXmlDataFromDataSourceResponse** element.

3.1.4.23.2 Elements

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

Element	Description
GetXmlDataFromDataSource	A request to initiate a **GetXmlDataFromDataSourceSoapIn** operation on the protocol server.
GetXmlDataFromDataSourceResponse	A response from the protocol server at completion of the **GetXmlDataFromDataSourceSoapOut** operation.

3.1.4.23.2.1 GetXmlDataFromDataSource

The **GetXmlDataFromDataSource** element takes a string that is implementation-specific XML and returns data usable for authoring that matches the schema of the data that would be returned by such an element during processing on the server. It is not necessary that the data be equal or that the data be obtained in exactly the same way as it would be during processing. However, the server MUST return data of the same schema so that the client can author constructs that use the data correctly. This element is defined as follows

```
<s:element name="GetXmlDataFromDataSource" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="queryXml" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

queryXml: An XML-encoded string that, after decoding, is a valid XML document containing implementation-specific<107> details on a query to run.

3.1.4.23.2.2 GetXmlDataFromDataSourceResponse

The **GetXmlDataFromDataSourceResponse** element is the response to
GetXmlDataFromDataSource. It represents data that the server returns to the client for
authoring. This data MUST match the schema of data that would be returned during processing. This
element is defined as follows:

```
<s:element name="GetXmlDataFromDataSourceResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="GetXmlDataFromDataSourceResult" type="s:string" minOccurs="1"
maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

GetXmlDataFromDataSourceResult: The XML-encoded string data returned for authoring. The
client MUST decode the string. If the result is valid XML, the client MUST treat this as a success
case, and the XML MUST be interpreted as the data to use for authoring. If it is not valid XML, the
client MUST treat this as a failure case.

3.1.4.23.3 Complex Types

None.

3.1.4.23.4 Simple Types

None.

3.1.4.23.5 Attributes

None.

3.1.4.23.6 Groups

None.

3.1.4.23.7 Attribute Groups

None.

3.1.4.24 RemoveWorkflowAssociation

The **RemoveWorkflowAssociation** operation<108> removes an association of a workflow from a
list (1). It undoes the **AssociateWorkflowMarkup** operation. This operation is defined as follows:

```
<wsdl:operation name="RemoveWorkflowAssociation"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="RemoveWorkflowAssociationSoapIn" />
   <wsdl:output message="RemoveWorkflowAssociationSoapOut" />
</wsdl:operation>
```

The client sends a **RemoveWorkflowAssociationSoapIn** request message, and the server responds with a **RemoveWorkflowAssociationSoapOut** response message.

3.1.4.24.1 Messages

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

Message	Description
RemoveWorkflowAssociationSoapIn	A request to initiate a **RemoveWorkflowAssociation** operation on the protocol server.
RemoveWorkflowAssociationSoapOut	A response from the protocol server at completion of the **RemoveWorkflowAssociation** operation.

3.1.4.24.1.1 RemoveWorkflowAssociationSoapIn

The **RemoveWorkflowAssociationSoapIn** message is the request message for **RemoveWorkflowAssociation**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/RemoveWorkflowAssociation
```

The SOAP Body contains a **RemoveWorkflowAssociation** element.

3.1.4.24.1.2 RemoveWorkflowAssociationSoapOut

The **RemoveWorkflowAssociationSoapOut** message is the response message for **RemoveWorkflowAssociation**.

The SOAP Body contains a **RemoveWorkflowAssociationResponse** element.

3.1.4.24.2 Elements

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

Element	Description
RemoveWorkflowAssociation	A request to initiate a **RemoveWorkflowAssociationSoapIn** operation on the protocol server.
RemoveWorkflowAssociationResponse	A response from the protocol server at completion of the **RemoveWorkflowAssociationSoapOut** operation.

3.1.4.24.2.1 RemoveWorkflowAssociation

The **RemoveWorkflowAssociation** element is defined as follows:

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

```
      <s:sequence>
         <s:element name="configUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="configVersion" minOccurs="0" maxOccurs="1">
            <s:simpleType>
               <s:restriction base="s:string">
                  <s:pattern value="[v|V]\d+\.\d+" />
               </s:restriction>
            </s:simpleType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

configUrl: An unencoded site-relative URL that indicates the location of the workflow configuration file for the workflow (2) to disassociate. The URL MUST NOT begin with a slash (/).

configVersion: A string prepended by a 'v' character identifying the displayed version of the workflow configuration file for which the association is to be removed. MUST be empty or omitted, or a valid version number recognized by the server. If it is empty or omitted, the latest version of the file is used.

3.1.4.24.2.2   RemoveWorkflowAssociationResponse

The **RemoveWorkflowAssociationResponse** element MUST be returned by the server in response to **RemoveWorkflowAssociation**. This element is defined as follows:

```
<s:element name="RemoveWorkflowAssociationResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="RemoveWorkflowAssociationResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

RemoveWorkflowAssociationResult: MUST contain an XML-encoded string that, when decoded, is either a non-negative integer indicating how many associations were successfully removed, one **DeclarativeWorkflowDisabled** element, or one **Error** element. It is possible for the operation to be successful and still return zero.

DeclarativeWorkflowDisabled: Indicates that the operation failed because an administrator has disabled declarative workflow (2) features on the server. When this element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="DeclarativeWorkflowDisabled" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema" />
```

Error: Indicates that the disassociation operation failed for a reason other than disabling of the workflow (2) feature. When this element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="Error" type="s:string" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema" />
```

When decoded, this element MUST contain a localized error string suitable for display to the user.

3.1.4.24.3 Complex Types

None.

3.1.4.24.4 Simple Types

None.

3.1.4.24.5 Attributes

None.

3.1.4.24.6 Groups

None.

3.1.4.24.7 Attribute Groups

None.

3.1.4.25 RenderWebPartForEdit

The **RenderWebPartForEdit** operation retrieves implementation-specific markup for a single Web Part that can be used to design pages. This operation is defined as follows:

```
<wsdl:operation name="RenderWebPartForEdit" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="RenderWebPartForEditSoapIn" />
   <wsdl:output message="RenderWebPartForEditSoapOut" />
</wsdl:operation>
```

The client sends a **RenderWebPartForEditSoapIn** request message, and the server responds with a **RenderWebPartForEditSoapOut** response message.

3.1.4.25.1 Messages

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

Message	Description
RenderWebPartForEditSoapIn	A request to initiate a **RenderWebPartForEdit** operation on the protocol server.
RenderWebPartForEditSoapOut	A response from the protocol server at completion of the **RenderWebPartForEdit** operation.

3.1.4.25.1.1 RenderWebPartForEditSoapIn

The **RenderWebPartForEditSoapIn** message is the request message for **RenderWebPartForEdit**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/RenderWebPartForEdit
```

The SOAP Body contains a **RenderWebPartForEdit** element.

3.1.4.25.1.2 RenderWebPartForEditSoapOut

The **RenderWebPartForEditSoapOut** message is the response message for **RenderWebPartForEdit**.

The SOAP Body contains a **RenderWebPartForEditResponse** element.

3.1.4.25.2 Elements

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

Element	Description
RenderWebPartForEdit	A request to initiate a **RenderWebPartForEditSoapIn** operation on the protocol server.
RenderWebPartForEditResponse	A response from the protocol server at completion of the **RenderWebPartForEditSoapOut** operation.

3.1.4.25.2.1 RenderWebPartForEdit

The **RenderWebPartForEdit** element is defined as follows:

```
<s:element name="RenderWebPartForEdit" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

webPartXml: An XML-encoded string that, when decoded, is in the DWP file format defined in **wpv2:WebPart** (section 2.2.4.2).

3.1.4.25.2.2 RenderWebPartForEditResponse

The **RenderWebPartForEditResponse** element MUST be returned by the server in response to **RenderWebPartForEdit**. This element is defined as follows:

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

```
            <s:element name="RenderWebPartForEditResult" type="s:string" minOccurs="1"
  maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
```

RenderWebPartForEditResult: An XML-encoded string that, when decoded, produces XML as follows:

```
<s:element name="Results" xmlns:s="http://www.w3.org/2001/XMLSchema">
    <s:complexType>
        <s:sequence>
            <s:element name="Result">
                <s:complexType>
                    <s:sequence>
                        <s:element name="DesignerInfo">
                            <s:complexType>
                                <s:sequence>
                                    <s:element name="Properties" type="s:string" minOccurs="1"
  maxOccurs="1"/>
                                    <s:element name="Preview" type="s:string" minOccurs="1"
  maxOccurs="1"/>
                                </s:sequence>
                            </s:complexType>
                        </s:element>
                    </s:sequence>
                    <s:attribute name="ID" type="s:string" use="required"/>
                    <s:attribute name="Code" type="s:int" use="required"/>
                    <s:attribute name="__error" type="s:string" use="optional"/>
                </s:complexType>
            </s:element>
        </s:sequence>
    </s:complexType>
</s:element>
```

Results: A container element for the **Result** element.

Results.Result: A container element for the **DesignerInfo** element.

Results.Result.DesignerInfo: A container element for the **Properties** and the **Preview** elements.

Results.Result.DesignerInfo.Properties: An XML-encoded string that, when decoded, is implementation-specific<109> markup containing properties of the Web Part that would be useful for the client application.

Results.Result.DesignerInfo.Preview: An XML-encoded string that, when decoded, is implementation-specific<110> markup suitable for display in the client application as a placeholder for the Web Part.

Results.Result.ID: MUST be set to "0,RenderForEdit".

Results.Result.Code: MUST be set to zero.

Results.Result.__error: If an error occurred during the operation, this attribute MUST contain a non-empty string describing the error. Otherwise, the attribute MUST either be omitted or an empty string.

3.1.4.25.3 Complex Types

None.

3.1.4.25.4 Simple Types

None.

3.1.4.25.5 Attributes

None.

3.1.4.25.6 Groups

None.

3.1.4.25.7 Attribute Groups

None.

3.1.4.26 SaveWebPart

The **SaveWebPart** operation updates a Web Part in a Web Part zone in a page. This operation is defined as follows:

```
<wsdl:operation name="SaveWebPart" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="SaveWebPartSoapIn" />
   <wsdl:output message="SaveWebPartSoapOut" />
</wsdl:operation>
```

The client sends a **SaveWebPartSoapIn** request message, and the server responds with a **SaveWebPartSoapOut** response message.

3.1.4.26.1 Messages

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

Message	Description
SaveWebPartSoapIn	A request to initiate a **SaveWebPart** operation on the protocol server.
SaveWebPartSoapOut	A response from the protocol server at completion of the **SaveWebPart** operation.

3.1.4.26.1.1 SaveWebPartSoapIn

The **SaveWebPartSoapIn** message is the request message for **SaveWebPart**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/SaveWebPart
```

The SOAP Body contains a **SaveWebPart** element.

3.1.4.26.1.2 SaveWebPartSoapOut

The **SaveWebPartSoapOut** message is the response message for **SaveWebPart**.

The SOAP Body contains a **SaveWebPartResponse** element.

3.1.4.26.2 Elements

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

Element	Description
SaveWebPart	A request to initiate a **SaveWebPartSoapIn** operation on the protocol server.
SaveWebPartResponse	A response from the protocol server at completion of the **SaveWebPartSoapOut** operation.

3.1.4.26.2.1 SaveWebPart

The **SaveWebPart** element updates an existing Web Part in a page. This element is defined as follows:

```
<s:element name="SaveWebPart" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

pageUrl: The location of the page. MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection.

storageKey: A GUID that specifies which Web Part to update. This MUST correspond to a Web Part that is specified in one of the Web Part zones in the view specified by the **storage** parameter. The Web Part MUST also be serializable in one of the formats supported by this operation.

webPartXml: An XML-encoded string containing the contents of the Web Part being saved. After decoding, this MUST be in a format<111> supported by the server.

storage: Specifies whether to save the Web Part referred to by the **storageKey** property to the public view or the current user's personal view of the page. If not set, the server MUST interpret it as "Personal".

3.1.4.26.2.2 SaveWebPartResponse

The **SaveWebPartResponse** element is the response to **SaveWebPart**.

If the call to **SaveWebPart** would change the underlying object type in the object-oriented system of the Web Part, the server MUST raise an error. Otherwise, the server returns the following:

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

3.1.4.26.3 Complex Types

None.

3.1.4.26.4 Simple Types

None.

3.1.4.26.5 Attributes

None.

3.1.4.26.6 Groups

None.

3.1.4.26.7 Attribute Groups

None.

3.1.4.27 SaveWebPart2

The **SaveWebPart2** operation<112> updates an existing Web Part in a page. In addition to the functionality of **SaveWebPart**, it enables the client to change the underlying object type of the Web Part if the parameter **allowTypeChange** is set to TRUE. This operation is defined as follows:

```
<wsdl:operation name="SaveWebPart2" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="SaveWebPart2SoapIn" />
   <wsdl:output message="SaveWebPart2SoapOut" />
</wsdl:operation>
```

The client sends a **SaveWebPart2SoapIn** request message, and the server responds with a **SaveWebPart2SoapOut** response message.

3.1.4.27.1 Messages

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

Message	Description
SaveWebPart2SoapIn	A request to initiate a **SaveWebPart2** operation on the protocol server.
SaveWebPart2SoapOut	A response from the protocol server at completion of the **SaveWebPart2** operation.

3.1.4.27.1.1 SaveWebPart2SoapIn

The **SaveWebPart2SoapIn** message is the request message for **SaveWebPart2**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/SaveWebPart2
```

The SOAP Body contains a **SaveWebPart2** element.

3.1.4.27.1.2 SaveWebPart2SoapOut

The **SaveWebPart2SoapOut** message is the response message for **SaveWebPart2**.

The SOAP Body contains a **SaveWebPart2Response** element.

3.1.4.27.2 Elements

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

Element	Description
SaveWebPart2	A request to initiate a **SaveWebPart2SoapIn** operation on the protocol server.
SaveWebPart2Response	A response from the protocol server at completion of the **SaveWebPart2SoapOut** operation.

3.1.4.27.2.1 SaveWebPart2

The **SaveWebPart2** element updates an existing Web Part in a page. This element is defined as follows:

```
<s:element name="SaveWebPart2" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1" />
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1" />
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1" />
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1" />
         <s:element name="allowTypeChange" type="s:boolean" minOccurs="0"  maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

pageUrl: The location of the page. MUST be formatted as either an absolute URL, a server-relative URL, or a site-relative URL. The location specified MUST be in the current site (2) or any child site (2) of the current site (2) that is also in the current site collection.

storageKey: A GUID that specifies which Web Part to update. MUST correspond to a Web Part that is specified one of the Web Part zones in the view specified by the **storage** parameter. The Web Part MUST also be serializable in one of the formats supported by this operation.

webPartXml: An XML-encoded string. After decoding, this MUST be in a format<113> supported by the server. This is the contents of the Web Part being saved.

storage: Specifies whether to save the Web Part referred to by the **storageKey** property to the public view or the current user's personal view of the page. If not set, the server MUST interpret it as "Personal".

allowTypeChange: If set to FALSE, the server MUST raise an error if the call to **SaveWebPart2** specifies that the underlying object type of the Web Part changes.

3.1.4.27.2.2 SaveWebPart2Response

The **SaveWebPart2Response** element is the response to **SaveWebPart2**. This element is defined as follows:

```
<s:element name="SaveWebPart2Response" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType/>
</s:element>
```

3.1.4.27.3 Complex Types

None.

3.1.4.27.4 Simple Types

None.

3.1.4.27.5 Attributes

None.

3.1.4.27.6 Groups

None.

3.1.4.27.7 Attribute Groups

None.

3.1.4.28 ValidateWorkflowMarkupAndCreateSupportObjects

The **ValidateWorkflowMarkupAndCreateSupportObjects** operation<114> causes the server to check a group of related workflow (2) files referenced by a workflow configuration file for validity and correctness and return error information if any are found to be invalid. This process depends on the workflow (2) implementation that the server provides. This operation is defined as follows:

```
<wsdl:operation name="ValidateWorkflowMarkupAndCreateSupportObjects"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:input message="ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn" />
   <wsdl:output message="ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut" />
</wsdl:operation>
```

The client sends a **ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn** request message, and the server responds with a **ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut** response message.

3.1.4.28.1 Messages

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

Message	Description
ValidateWorkflowMarkupAndCreateSupportObjects SoapIn	A request to initiate a **ValidateWorkflowMarkupAndCreateSupport Objects** operation on the protocol server.
ValidateWorkflowMarkupAndCreateSupportObjects SoapOut	A response from the protocol server at completion of the **ValidateWorkflowMarkupAndCreateSupport Objects** operation.

3.1.4.28.1.1 ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn

The **ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn** message is the request message for **ValidateWorkflowMarkupAndCreateSupportObjects**.

The SOAP Action value of the message is defined as follows:

```
http://microsoft.com/sharepoint/webpartpages/ValidateWorkflowMarkupAndCreateSupportObjects
```

The SOAP Body contains a **ValidateWorkflowMarkupAndCreateSupportObjects** element.

3.1.4.28.1.2 ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut

The **ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut** message is the response message for **ValidateWorkflowMarkupAndCreateSupportObjects**.

The SOAP Body contains a **ValidateWorkflowMarkupAndCreateSupportObjectsResponse** element.

3.1.4.28.2 Elements

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

Element	Description
ValidateWorkflowMarkupAndCreateSupportObje cts	A request to initiate a **ValidateWorkflowMarkupAndCreateSupportObj ectsSoapIn** operation on the protocol server.
ValidateWorkflowMarkupAndCreateSupportObje ctsResponse	A response from the protocol server at completion of the **ValidateWorkflowMarkupAndCreateSupportObj ectsSoapOut** operation.

3.1.4.28.2.1 ValidateWorkflowMarkupAndCreateSupportObjects

The **ValidateWorkflowMarkupAndCreateSupportObjects** element is defined as follows:

```
<s:element name="ValidateWorkflowMarkupAndCreateSupportObjects"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="workflowMarkupText" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="rulesText" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="configBlob" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="flag" type="s:int" minOccurs="1" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
```

workflowMarkupText: MUST contain XML-encoded workflow (2) markup to validate, in an implementation-specific<115> format.

rulesText: If not empty, contains XML-encoded markup in the format of a workflow rules file to validate. The element MUST be present, but if it is empty, the server SHOULD assume that the workflow (2) does not have any rules to validate. The format of the contents is implementation-specific.<116>

configBlob: If not empty, contains XML-encoded markup in the format of a workflow configuration file to validate. The element MUST be present, but the contents MAY be empty. The format of the contents is implementation-specific.<117>

flag: MAY<118> contain implementation-specific flags to control the behavior of the server.

3.1.4.28.2.2 ValidateWorkflowMarkupAndCreateSupportObjectsResponse

The **ValidateWorkflowMarkupAndCreateSupportObjectsResponse** element MUST be returned by the server in response to **ValidateWorkflowMarkupAndCreateSupportObjects**. This element is defined as follows:

```
<s:element name="ValidateWorkflowMarkupAndCreateSupportObjectsResponse"
xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="ValidateWorkflowMarkupAndCreateSupportObjectsResult"
type="s:string" minOccurs="1" maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
```

ValidateWorkflowMarkupAndCreateSupportObjectsResult: MUST contain an XML-encoded string that, when decoded, is either one **DeclarativeWorkflowDisabled** element, one **Success** element, or one **Error** element, defined as follows:

DeclarativeWorkflowDisabled: Indicates that the operation failed because an administrator has disabled declarative workflow (2) features on the server. When this element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="DeclarativeWorkflowDisabled" minOccurs="0" maxOccurs="1"
xmlns:s="http://www.w3.org/2001/XMLSchema" />
```

Success: Indicates that the validation operation was successful. When this element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="Success" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType>
      <s:sequence>
         <s:element name="TaskList">
            <s:complexType>
               <s:attribute name="ID" type="s1:guid" use="required"/>
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

Success.TaskList: Provides information about the list (1) of tasks that the workflow (2) is to be assigned.

Success.TaskList.ID: A string that uniquely identifies the list (1) of tasks.<119>

Error: Indicates that the validation operation failed for a reason other than disabling of the workflow (2) feature. The **Error** element MUST have either one **CompilerError** element for each compilation error that occurred or one localized error string suitable for display to the user. When the **Error** element is present, it MUST be an XML-encoded string that conforms to this schema when decoded, as follows:

```
<s:element name="Error" xmlns:s="http://www.w3.org/2001/XMLSchema">
   <s:complexType mixed="true">
      <s:sequence minOccurs="0" maxOccurs="unbounded">
         <s:element name="CompilerError">
            <s:complexType>
               <s:attribute name="Line" type="s:int" use="required"/>
               <s:attribute name="Column" type="s:int" use="required"/>
               <s:attribute name="Text" type="s:string" use="required"/>
            </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
```

Error.CompilerError: Provides information about one error that was encountered while validating the workflow markup, workflow rules, or workflow configuration file text.

Error.CompilerError.Line: The one-based line number on which the compilation error was found.

Error.CompilerError.Column: The one-based character index of the line on which the compilation error was found.

Error.CompilerError.Text: The message associated with the compilation error.

Error.*: If the **Error** element contains only a text value, the client MUST interpret it as a localized error string suitable for display to a user.

---

143 / 224

3.1.4.28.3   Complex Types

None.

3.1.4.28.4   Simple Types

None.

3.1.4.28.5   Attributes

None.

3.1.4.28.6   Groups

None.

3.1.4.28.7   Attribute Groups

None.

3.1.5   Timer Events

None.

3.1.6   Other Local Events

None.

4   Protocol Examples

4.1   XML-encoded XML

Many of the operations specified in this document return, or take as parameters, XML markup that has itself been XML-encoded. Generally, the receiver of the message takes the text contents of an XML element as a string, runs the XML decode operation on that string to convert character entities into special XML characters such as angle brackets, and then parses the decoded string as XML. The following is an example of XML-encoded XML:

```
<Result>
  &lt;Error&gt;There was an unspecified error.&lt;/Error&gt;
</Result>
```

In this example, the XML **Result** contains text content. That text content is itself XML, but it is encoded, and thus has to be decoded before it can be parsed. The XML decode operation replaces "<" with "<", and ">" with ">". So, after the operation, the text content of the **Result** element is as follows:

```
<Error>There was an unspecified error.</Error>
```

Now, the receiver of the message can parse the result as XML.

4.2   Workflow

This protocol includes several operations that can be performed on workflows (2). Although they can be called in any order, the requirements of the operations result in a fairly specific sequence that is generally followed when a client application enables a user to create or edit a workflow (2).
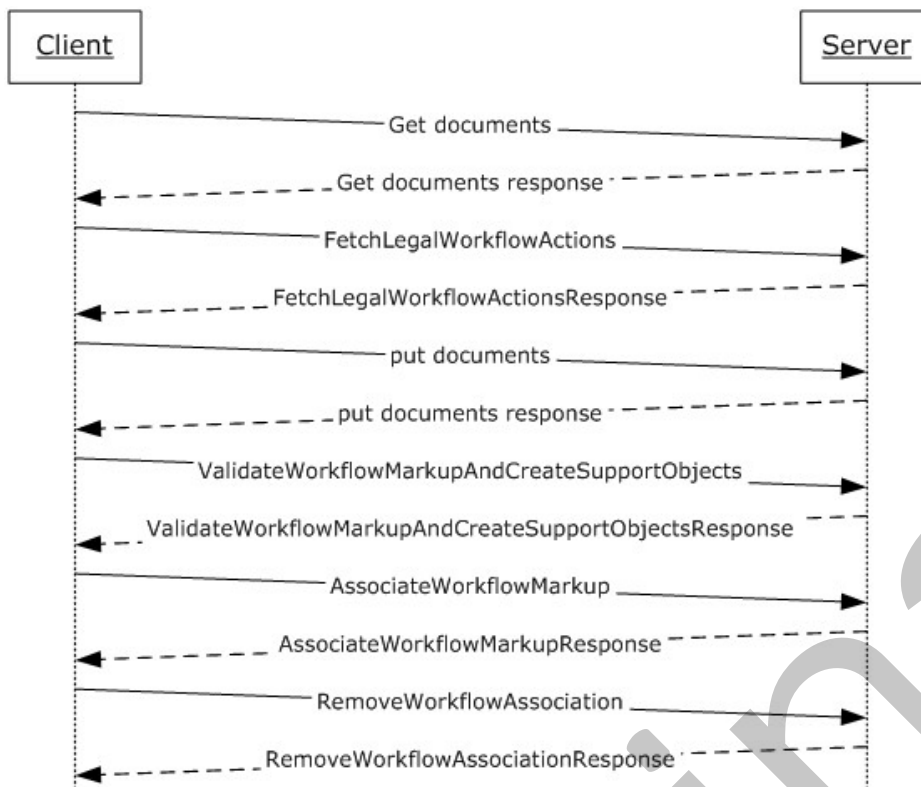
Figure 2: Workflow authoring

4.2.1   Retrieve the Workflow Files

If the client application is opening an existing saved workflow (2), it needs copies of the necessary workflow (2) files. To download them from the server, it uses the FrontPage Server Extensions (FPSE) Remote Protocol **get document** method, as described in [MS-FPSE] section 3.1.5.3.6 or the **get documents** method, as described in [MS-FPSE] section 3.1.5.3.7. It downloads the workflow configuration file, workflow markup file, and workflow rules file (if present), as well as any other server implementation-specific files that it requires.

It is not required to use the FPSE protocol to download and upload files. Other operations could be used, depending on which protocols are supported by the server.

4.2.2   Retrieve User Interface and Localization Information

After opening the existing workflow (2) files, or creating new ones as applicable, the client application needs to know how to display elements of the workflow (conditions and actions (2)) in its UI. To get this information, it uses the **FetchLegalWorkflowActions** operation on this protocol.

A sample request is similar to the following:

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

146 / 224

```
            </FetchLegalWorkflowActions>
        </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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><FetchLegalWorkflowActionsResponse
xmlns="http://microsoft.com/sharepoint/webpartpages">
        <FetchLegalWorkflowActionsResult>&lt;?xml version="1.0" encoding="utf-
8"?&gt;&lt;WorkflowInfo Language="en-us"&gt;&lt;Conditions And="and" Or="or" Not="not"
When="If" Else="Else if"&gt;&lt;Default
ItemKeyType="Microsoft.SharePoint.Workflow.SPItemKey"&gt;&lt;RuleDesigner Sentence="%1 %2
%3"&gt;&lt;FieldBind Id="1" Field="left" Text="value" DesignerType="TextBox"
/&gt;&lt;FieldBind Id="2" Field="operator" DesignerType="Operator" OperatorTypeFrom="left"
Text="this test"&gt;&lt;Option Name="equals" Value="Equal" /&gt;&lt;Option Name="not equals"
Value="NotEqual" /&gt;&lt;Option Name="is greater than" Value="GreaterThan"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is greater than or equal to" Value="GreaterThanOrEqual"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is less than" Value="LessThan"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is less than or equal to" Value="LessThanOrEqual"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is empty" Value="IsEmpty" TypeFilter="System.String" UnaryHides="right" /&gt;&lt;Option
Name="is not empty" Value="NotIsEmpty" TypeFilter="System.String" UnaryHides="right"
/&gt;&lt;Option Name="begins with" Value="StartsWith" TypeFilter="System.String"
/&gt;&lt;Option Name="does not begin with" Value="NotStartsWith" TypeFilter="System.String"
/&gt;&lt;Option Name="ends with" Value="EndsWith" TypeFilter="System.String" /&gt;&lt;Option
Name="does not end with" Value="NotEndsWith" TypeFilter="System.String" /&gt;&lt;Option
Name="contains" Value="Contains" TypeFilter="System.String" /&gt;&lt;Option Name="does not
contain" Value="NotContains" TypeFilter="System.String" /&gt;&lt;Option Name="matches regular
expression" Value="Matches" TypeFilter="System.String" /&gt;&lt;Option Name="equals (ignoring
case)" Value="EqualNoCase" TypeFilter="System.String" /&gt;&lt;Option Name="contains
(ignoring case)" Value="ContainsNoCase" TypeFilter="System.String" /&gt;&lt;Option
Name="equals (ignoring time)" Value="EqualNoTime" TypeFilter="System.DateTime"
/&gt;&lt;/FieldBind&gt;&lt;FieldBind Id="3" Field="right" Text="value" TypeFrom="left"
DesignerType="TextBox" /&gt;&lt;/RuleDesigner&gt;&lt;/Default&gt;&lt;Condition Name="If
current item field equals value" AppliesTo="list" UsesCurrentItem="true"
ItemKeyType="Microsoft.SharePoint.Workflow.SPItemKey"&gt;&lt;RuleDesigner Sentence="%1 %2
%3"&gt;&lt;FieldBind Id="1" Field="left" DesignerType="FieldNames" Text="field"
/&gt;&lt;FieldBind Id="2" Field="operator" DesignerType="Operator" OperatorTypeFrom="left"
Text="this test"&gt;&lt;Option Name="equals" Value="Equal" /&gt;&lt;Option Name="not equals"
Value="NotEqual" /&gt;&lt;Option Name="is greater than" Value="GreaterThan"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is greater than or equal to" Value="GreaterThanOrEqual"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is less than" Value="LessThan"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is less than or equal to" Value="LessThanOrEqual"
TypeFilter="System.Double;System.Int32;System.Uint32;System.DateTime" /&gt;&lt;Option
Name="is empty" Value="IsEmpty" TypeFilter="System.String" UnaryHides="right" /&gt;&lt;Option
Name="is not empty" Value="NotIsEmpty" TypeFilter="System.String" UnaryHides="right"
/&gt;&lt;Option Name="begins with" Value="StartsWith" TypeFilter="System.String"
/&gt;&lt;Option Name="does not begin with" Value="NotStartsWith" TypeFilter="System.String"
/&gt;&lt;Option Name="ends with" Value="EndsWith" TypeFilter="System.String" /&gt;&lt;Option
Name="does not end with" Value="NotEndsWith" TypeFilter="System.String" /&gt;&lt;Option
Name="contains" Value="Contains" TypeFilter="System.String" /&gt;&lt;Option Name="does not
contain" Value="NotContains" TypeFilter="System.String" /&gt;&lt;Option Name="matches regular
expression" Value="Matches" TypeFilter="System.String" /&gt;&lt;Option Name="equals (ignoring
case)" Value="EqualNoCase" TypeFilter="System.String" /&gt;&lt;Option Name="contains
```

```
        ClassName="Microsoft.SharePoint.WorkflowActions.ImpersonationSequenceActivity"
        Assembly="Microsoft.SharePoint.WorkflowActions, Version=14.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c" AppliesTo="all" InvocationTags="ImpersonationSequence"
        ProhibitedInvocationTags="ImpersonationSequence"
        /&gt;&lt;/CompositeSteps&gt;&lt;VariableTypes&gt;&lt;VariableType Type="System.Boolean"
        DisplayName="Boolean"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceBooleanActivity"
        /&gt;&lt;VariableType Type="System.DateTime" DisplayName="Date/Time"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceDateTimeActivity"
        /&gt;&lt;VariableType Type="System.Double" DisplayName="Number"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceDoubleActivity"
        /&gt;&lt;VariableType Type="System.Int32" DisplayName="Integer"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceIntActivity"
        /&gt;&lt;VariableType Type="System.String" Default="true" DisplayName="String"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceStringActivity"
        /&gt;&lt;VariableType Type="Microsoft.SharePoint.Workflow.SPItemKey" DisplayName="List Item
        Id" Assembly="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c"
        CoercionActivity="Microsoft.SharePoint.WorkflowActions.CoerceSPItemKeyActivity"
        ItemIdType="true" /&gt;&lt;/VariableTypes&gt;&lt;AssemblyRedirects&gt;&lt;AssemblyRedirect
        From="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c" To="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c" /&gt;&lt;AssemblyRedirect From="Microsoft.SharePoint,
        Version=12.0.0.0, Culture=neutral, PublicKeyToken=null" To="Microsoft.SharePoint,
        Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" /&gt;&lt;AssemblyRedirect
        From="Microsoft.SharePoint.WorkflowActions, Version=12.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c" To="Microsoft.SharePoint.WorkflowActions, Version=14.0.0.0,
        Culture=neutral, PublicKeyToken=71e9bce111e9429c" /&gt;&lt;AssemblyRedirect
        From="Microsoft.SharePoint.WorkflowActions, Version=12.0.0.0, Culture=neutral,
        PublicKeyToken=null" To="Microsoft.SharePoint.WorkflowActions, Version=14.0.0.0,
        Culture=neutral, PublicKeyToken=71e9bce111e9429c"
        /&gt;&lt;/AssemblyRedirects&gt;&lt;Coercions&gt;&lt;!-- lookup related --&gt;&lt;Coercion
        Name="LookupColumnDataOnlyAsDateTime" Assembly="Microsoft.SharePoint.WorkflowActions,
        Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" DisplayName="Lookup Value
        (as Date/Time)" CatchAll="false" RankDefault="51" SourceType="System.String"
        SourceFieldType="Lookup" DestinationType="System.DateTime" /&gt;&lt;Coercion
        Name="LookupColumnDataOnlyAsDouble" Assembly="Microsoft.SharePoint.WorkflowActions,
        Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" DisplayName="Lookup Value
        (as Number)" CatchAll="false" RankDefault="52" SourceType="System.String"
        SourceFieldType="Lookup" DestinationType="System.Double" /&gt;&lt;Coercion Name="AsDNString"
        Assembly="Microsoft.SharePoint.WorkflowActions, Version=14.0.0.0, Culture=neutral,
        PublicKeyToken=71e9bce111e9429c" DisplayName="As String" CatchAll="true" RankDefault="1"
        DestinationType="System.String"
        /&gt;&lt;/Coercions&gt;&lt;/WorkflowInfo&gt;</FetchLegalWorkflowActionsResult>
              </FetchLegalWorkflowActionsResponse>
          </soap:Body>
      </soap:Envelope>
```

4.2.3   Upload the Workflow to the Server

Once the user has made changes to the workflow (2) and is ready to save them, the client
application uploads the workflow (2) files to the server using the FPSE **put document** method, as
described in [MS-FPSE] section 3.1.5.3.11 or the **put documents** method, as described in [MS-
FPSE] section 3.1.5.3.12. The location to which workflow (2) files are saved is server
implementation-specific.

4.2.4   Validate and Compile the Workflow

After the updated workflow (2) files are uploaded to the server, the client validates and compiles
them by calling the **ValidateWorkflowMarkupAndCreateSupportObjects** operation on this
protocol, passing it the URL of the workflow configuration file that was just uploaded and a server

implementation-specific *flag* parameter. (The server implementation could require that this operation be called more than once with different values for the *flag* parameter.)

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <ValidateWorkflowMarkupAndCreateSupportObjects
xmlns="http://microsoft.com/sharepoint/webpartpages">
        <workflowMarkupText>&lt;ns0:RootWorkflowActivityWithData
x:Class=&quot;Microsoft.SharePoint.Workflow.ROOT&quot; x:Name=&quot;ROOT&quot;
xmlns=&quot;http://schemas.microsoft.com/winfx/2006/xaml/workflow&quot;
xmlns:x=&quot;http://schemas.microsoft.com/winfx/2006/xaml&quot; xmlns:ns0=&quot;clr-
namespace:Microsoft.SharePoint.WorkflowActions;Assembly=Microsoft.SharePoint.WorkflowActions,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=null&quot;&gt;
            &lt;ns0:RootWorkflowActivityWithData.WorkflowFields&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__list&quot; Type=&quot;System.String&quot;
/&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__item&quot;
Type=&quot;Microsoft.SharePoint.Workflow.SPItemKey, Microsoft.SharePoint, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c&quot; /&gt;
             &lt;ns0:WorkflowDataField Name=&quot;__context&quot;
Type=&quot;Microsoft.SharePoint.WorkflowActions.WorkflowContext&quot; /&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__initParams&quot;
Type=&quot;Microsoft.SharePoint.Workflow.SPWorkflowActivationProperties&quot; /&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__workflowId&quot;
Type=&quot;System.Guid&quot; /&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__historylist&quot;
Type=&quot;System.String&quot; /&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__tasklist&quot;
Type=&quot;System.String&quot; /&gt;
            &lt;ns0:WorkflowDataField Name=&quot;__itemId&quot; Type=&quot;System.Int32&quot;
/&gt;
            &lt;/ns0:RootWorkflowActivityWithData.WorkflowFields&gt;
            &lt;ns0:OnWorkflowActivated WorkflowProperties=&quot;{ActivityBind
ROOT,Path=__initParams}&quot; x:Name=&quot;ID1&quot;&gt;
            &lt;ns0:OnWorkflowActivated.CorrelationToken&gt;
            &lt;wf0:CorrelationToken Name=&quot;refObject&quot;
OwnerActivityName=&quot;ROOT&quot;
xmlns:wf0=&quot;http://schemas.microsoft.com/winfx/2006/xaml/workflow&quot; /&gt;
            &lt;/ns0:OnWorkflowActivated.CorrelationToken&gt;
            &lt;/ns0:OnWorkflowActivated&gt;
            &lt;ns0:ApplyActivation __Context=&quot;{ActivityBind ROOT,Path=__context}&quot;
x:Name=&quot;ID2&quot; __WorkflowProperties=&quot;{ActivityBind ROOT,Path=__initParams}&quot;
/&gt;
            &lt;SequenceActivity x:Name=&quot;ID3&quot; Description=&quot;Step 1&quot; /&gt;
            &lt;/ns0:RootWorkflowActivityWithData&gt;</workflowMarkupText>
<rulesText></rulesText><configBlob>&lt;WorkflowConfig Version=&quot;14.0.0.4730&quot;&gt;
            &lt;Template BaseID=&quot;{0501C0DC-A90D-4464-B880-8C0A15D5ACED}&quot;
DocLibID=&quot;{6E0D3D35-7B8F-4CA7-A770-2BB09374209F}&quot; Category=&quot;List&quot;
Name=&quot;Simul-Flow&quot; Description=&quot;&quot;/&gt;
            &lt;Association ListID=&quot;{2843EA1B-EFE1-4FCC-A39B-A46121D3EFD6}&quot;
TaskListID=&quot;{6F387CC4-889C-4BCF-81DB-16683596A2DD}&quot;
StartManually=&quot;true&quot;/&gt;
            &lt;ContentTypes/&gt;
            &lt;Initiation URL=&quot;Workflows/Simul-Flow/Simul-Flow.aspx&quot;&gt;
            &lt;Fields/&gt;
            &lt;Parameters/&gt;
            &lt;DefaultData/&gt;
```

```
                   &lt;/Initiation&gt;
                   &lt;Extended&gt;
                   &lt;Fields/&gt;
                   &lt;/Extended&gt;
                   &lt;MetaData&gt;
                   &lt;ExtendedStatusColumnValues&gt;
                   &lt;StatusColumnValue&gt;Canceled&lt;/StatusColumnValue&gt;
                   &lt;StatusColumnValue&gt;Approved&lt;/StatusColumnValue&gt;
                   &lt;StatusColumnValue&gt;Rejected&lt;/StatusColumnValue&gt;
                   &lt;/ExtendedStatusColumnValues&gt;
                   &lt;/MetaData&gt;
                   &lt;/WorkflowConfig&gt;
               </configBlob>
               <flag>2</flag>
           </ValidateWorkflowMarkupAndCreateSupportObjects>
       </soap:Body>
   </soap:Envelope>
```

A sample response is similar to the following:

```
<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>
        <ValidateWorkflowMarkupAndCreateSupportObjectsResponse
xmlns="http://microsoft.com/sharepoint/webpartpages">
        <ValidateWorkflowMarkupAndCreateSupportObjectsResult>&lt;Success&gt;&lt;TaskList
ID="{6F387CC4-889C-4BCF-81DB-16683596A2DD}"
/&gt;&lt;/Success&gt;</ValidateWorkflowMarkupAndCreateSupportObjectsResult>
        </ValidateWorkflowMarkupAndCreateSupportObjectsResponse>
    </soap:Body>
</soap:Envelope>
```

4.2.5 Associate the Workflow to a List

Once the workflow (2) has been compiled, the next step is to associate it with a list (1) on the
server so that it can be run on items in the list (1). This is done with the
AssociateWorkflowMarkup operation, which also takes the URL of the workflow configuration file.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <AssociateWorkflowMarkup xmlns="http://microsoft.com/sharepoint/webpartpages">
        <configUrl> Example/Example.xml</configUrl>
        <configVersion>V1.0</configVersion>
        </AssociateWorkflowMarkup>
    </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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>
      <AssociateWorkflowMarkupResponse xmlns="http://microsoft.com/sharepoint/webpartpages">
         <AssociateWorkflowMarkupResult>&lt;Success&gt;0501c0dc-a90d-4464-b880-
8c0a15d5aced&lt;/Success&gt;</AssociateWorkflowMarkupResult>
      </AssociateWorkflowMarkupResponse>
   </soap:Body>
</soap:Envelope>
```

4.2.5.1 Remove a Workflow Association

To undo the changes made by **AssociateWorkflowMarkup**, the client application can use the
RemoveWorkflowAssocation operation.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <RemoveWorkflowAssociation xmlns="http://microsoft.com/sharepoint/webpartpages">
         <configUrl>Example/Example.xml</configUrl>
         <configVersion></configVersion>
      </RemoveWorkflowAssociation>
   </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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>
      <RemoveWorkflowAssociationResponse
xmlns="http://microsoft.com/sharepoint/webpartpages">
         <RemoveWorkflowAssociationResult>1</RemoveWorkflowAssociationResult>
      </RemoveWorkflowAssociationResponse>
   </soap:Body>
</soap:Envelope>
```

4.3 Web Part Authoring

This protocol contains several operations for manipulating Web Parts on a page.

4.3.1 Ensure That a Web Part Is Present in a Web Part Page

When an authoring client needs to ensure that a Web Part exists in a page, it obtains the list of
existing Web Parts, and if the intended Web Part does not exist in the page, adds such a Web Part
to the page.
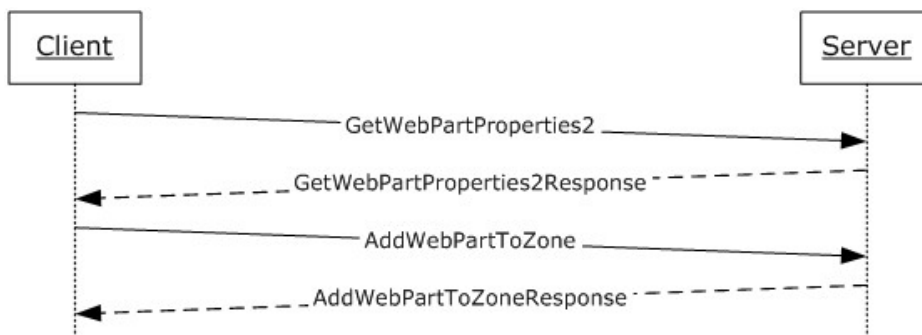
Figure 3: Ensuring a Web Part is present in a page

4.3.1.1 Get the List of Web Parts in a Web Part Page

To obtain the list of Web Parts in the page, the client can send a **GetWebPartProperties2** request to the server, and the server response **GetWebPartProperties2Response** will contain the list of Web Parts in all Web Part zones currently in the page.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <GetWebPartProperties2 xmlns="http://microsoft.com/sharepoint/webpartpages">
         <pageUrl>%2fdefault.aspx</pageUrl>
         <storage>Shared</storage>
         <behavior>Version3</behavior>
      </GetWebPartProperties2>
   </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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>
      <GetWebPartProperties2Response xmlns="http://microsoft.com/sharepoint/webpartpages">
         <GetWebPartProperties2Result>
            <WebParts>
               <WebPart ID="a4439cb8-de06-4f59-8698-4296fb3d7955">
                  <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
                     <metaData>
                        <type name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" />
                        <importErrorMessage>Cannot import this Web Part.</importErrorMessage>
                     </metaData>
                     <data>
                     <properties>
                       <property name="InitialAsyncDataFetch" type="bool">False</property>
                       <property name="ChromeType" type="chrometype">Default</property>
                       <property name="Title" type="string" />
                       <property name="Height" type="string" />
```

154 / 224

```
                            <property name="CacheXslStorage" type="bool">True</property>
                            <property name="ListDisplayName" type="string" null="true" />
                            <property name="AllowZoneChange" type="bool">True</property>
                            <property name="AllowEdit" type="bool">True</property>
                            <property name="XmlDefinitionLink" type="string" />
                            <property name="DataFields" type="string" />
                            <property name="Hidden" type="bool">False</property>
                            <property name="ListName" type="string">{9C26F0DF-2200-4773-81BF-
   DD86F8649CB9}</property>
                            <property name="NoDefaultStyle" type="string" />
                            <property name="AutoRefresh" type="bool">False</property>
                            <property name="ViewFlag" type="string">8396813</property>
                            <property name="Direction" type="direction">NotSet</property>
                            <property name="AutoRefreshInterval" type="int">60</property>
                            <property name="AllowConnect" type="bool">True</property>
                            <property name="Description" type="string" />
                            <property name="AllowClose" type="bool">True</property>
                            <property name="ShowWithSampleData" type="bool">False</property>
                            <property name="ParameterBindings" type="string">
                                &lt;ParameterBinding Name="dvt_sortdir"
   Location="Postback;Connection"/&gt;
                                &lt;ParameterBinding Name="dvt_sortfield"
   Location="Postback;Connection"/&gt;
                                &lt;ParameterBinding Name="dvt_startposition" Location="Postback"
   DefaultValue=""/&gt;
                                &lt;ParameterBinding Name="dvt_firstrow"
   Location="Postback;Connection"/&gt;
                                &lt;ParameterBinding Name="OpenMenuKeyAccessible"
   Location="Resource(wss,OpenMenuKeyAccessible)" /&gt;
                                &lt;ParameterBinding Name="open_menu"
   Location="Resource(wss,open_menu)" /&gt;
                                &lt;ParameterBinding Name="select_deselect_all"
   Location="Resource(wss,select_deselect_all)" /&gt;
                                &lt;ParameterBinding Name="idPresEnabled"
   Location="Resource(wss,idPresEnabled)" /&gt;
                                &lt;ParameterBinding Name="NoAnnouncements"
   Location="Resource(wss,noevents_toaddeventclickaddnewevent)" /&gt;
                                &lt;ParameterBinding Name="NoAnnouncementsHowTo"
   Location="Resource(wss,noevents_toaddeventclickaddnewevent_howto)" /&gt;
                                &lt;ParameterBinding Name="AddNewAnnouncement"
   Location="Resource(wss,AddNewEvent)" /&gt;
                                &lt;ParameterBinding Name="MoreAnnouncements"
   Location="Resource(wss,more_events)" /&gt;
                            </property>
                            <property name="Xsl" type="string" null="true" />
                            <property name="CacheXslTimeOut" type="int">86400</property>
                            <property name="WebId" type="System.Guid, mscorlib, Version=2.0.0.0,
   Culture=neutral, PublicKeyToken=b77a5c561934e089">00000000-0000-0000-0000-
   000000000000</property>
                            <property name="ListUrl" type="string" null="true" />
                            <property name="DataSourceID" type="string" />
                            <property name="FireInitialRow" type="bool">True</property>
                            <property name="ManualRefresh" type="bool">False</property>
                            <property name="ViewFlags" type="Microsoft.SharePoint.SPViewFlags,
   Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
   PublicKeyToken=71e9bce111e9429c">Html, TabularView, Hidden, RecurrenceRowset,
   Mobile</property>
                            <property name="ChromeState" type="chromestate">Normal</property>
                            <property name="AllowHide" type="bool">True</property>
                            <property name="PageSize" type="int">-1</property>
                            <property name="SampleData" type="string" null="true" />
```

```
                              <property name="BaseXsltHashKey" type="string" null="true" />
                              <property name="AsyncRefresh" type="bool">False</property>
                              <property name="HelpMode" type="helpmode">Modeless</property>
                              <property name="ListId" type="System.Guid, mscorlib, Version=2.0.0.0,
  Culture=neutral, PublicKeyToken=b77a5c561934e089">9c26f0df-2200-4773-81bf-
  dd86f8649cb9</property>
                              <property name="DataSourceMode"
  type="Microsoft.SharePoint.WebControls.SPDataSourceMode, Microsoft.SharePoint,
  Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c">List</property>
                              <property name="AllowMinimize" type="bool">True</property>
                              <property name="TitleUrl" type="string">/Lists/Calendar</property>
                              <property name="CatalogIconImageUrl" type="string" />
                              <property name="DataSourcesString" type="string" />
                              <property name="GhostedXslLink" type="string">main.xsl</property>
                              <property name="PageType" type="Microsoft.SharePoint.PAGETYPE,
  Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
  PublicKeyToken=71e9bce111e9429c">PAGE_NORMALVIEW</property>
                              <property name="DisplayName" type="string" />
                              <property name="UseSQLDataSourcePaging" type="bool">True</property>
                              <property name="Width" type="string" />
                              <property name="ExportMode" type="exportmode">All</property>
                              <property name="XslLink" type="string" null="true" />
                              <property name="ViewContentTypeId" type="string" />
                              <property name="HelpUrl" type="string" />
                              <property name="XmlDefinition" type="string">&lt;View Name="{A4439CB8-
  DE06-4F59-8698-4296FB3D7955}" MobileView="TRUE" Type="HTML" Hidden="TRUE"
  RecurrenceRowset="TRUE" DisplayName="" Url="/default.aspx" Level="1" BaseViewID="0"
  ContentTypeID="0x"&gt;
                                &lt;Query&gt;&lt;Where&gt;&lt;DateRangesOverlap&gt;&lt;
  FieldRef Name="EventDate"/&gt;
                                &lt;FieldRef Name="EndDate"/&gt;
                                &lt;FieldRef Name="RecurrenceID"/&gt;
                                &lt;Value Type="DateTime"&gt;&lt;Now/&gt;
                                &lt;/Value&gt;&lt;/DateRangesOverlap&gt;
                                  &lt;/Where&gt;&lt;/Query&gt;
                                  &lt;ViewFields&gt;&lt;FieldRef Name="EventDate"
  Explicit="TRUE"/&gt;
                                  &lt;FieldRef Name="LinkTitleNoMenu" Explicit="TRUE"/&gt;
                                  &lt;FieldRef Name="Description" Explicit="TRUE"/&gt;
                                  &lt;FieldRef Name="Attachments" Explicit="TRUE"/&gt;
                                  &lt;FieldRef Name="fRecurrence" Explicit="TRUE"/&gt;
                                  &lt;/ViewFields&gt;
                                  &lt;RowLimit&gt;10&lt;/RowLimit&gt;
                                  &lt;Toolbar Type="Freeform"/&gt;
                                  &lt;/View&gt;
                              </property>
                              <property name="Default" type="string" />
                              <property name="TitleIconImageUrl" type="string" />
                              <property name="MissingAssembly" type="string">Cannot import this
  Web Part.</property>
                              <property name="SelectParameters" type="string" />
                          </properties>
                      </data>
                  </webPart>
              </WebPart>
          </WebParts>
        </GetWebPartProperties2Result>
      </GetWebPartProperties2Response>
    </soap:Body>
```

```
        </soap:Envelope>
```

4.3.1.2   Add a Web Part to a Page

If the client determines that the intended Web Part does not exist in the page, it can call
AddWebPartToZone to add a new Web Part to the page. The server responds with
AddWebPartToZoneResponse, and the client can record the **storageKey** of the Web Part just
added.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <AddWebPartToZone xmlns="http://microsoft.com/sharepoint/webpartpages">
            <pageUrl>default.aspx</pageUrl>
            <webPartXml>&lt;webParts &gt;
            &lt;webPart  xmlns=&quot;http://schemas.microsoft.com/WebPart/v3&quot;    &gt;
            &lt;metaData&gt;
            &lt;type name=&quot;Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c&quot; /&gt;
            &lt;importErrorMessage&gt;Cannot import this Web Part.&lt;/importErrorMessage&gt;
            &lt;/metaData&gt;
            &lt;data&gt;
            &lt;properties&gt;
            &lt;property name=&quot;InitialAsyncDataFetch&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
            &lt;property name=&quot;ChromeType&quot;
type=&quot;chrometype&quot;&gt;Default&lt;/property&gt;
            &lt;property name=&quot;Title&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;Height&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;CacheXslStorage&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
            &lt;property name=&quot;ListDisplayName&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
            &lt;property name=&quot;AllowZoneChange&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
            &lt;property name=&quot;AllowEdit&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
            &lt;property name=&quot;XmlDefinitionLink&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;DataFields&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;Hidden&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
            &lt;property name=&quot;ListName&quot; type=&quot;string&quot;&gt;{2843EA1B-EFE1-
4FCC-A39B-A46121D3EFD6}&lt;/property&gt;
            &lt;property name=&quot;NoDefaultStyle&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;AutoRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
            &lt;property name=&quot;ViewFlag&quot;
type=&quot;string&quot;&gt;8388621&lt;/property&gt;
            &lt;property name=&quot;Direction&quot;
type=&quot;direction&quot;&gt;NotSet&lt;/property&gt;
            &lt;property name=&quot;AutoRefreshInterval&quot;
type=&quot;int&quot;&gt;60&lt;/property&gt;
            &lt;property name=&quot;AllowConnect&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
```

```
&lt;property name=&quot;Description&quot; type=&quot;string&quot; /&gt;
&lt;property name=&quot;AllowClose&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
&lt;property name=&quot;ShowWithSampleData&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
&lt;property name=&quot;ParameterBindings&quot; type=&quot;string&quot;&gt;
&amp;lt;ParameterBinding Name=&quot;dvt_sortdir&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
&amp;lt;ParameterBinding Name=&quot;dvt_sortfield&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
&amp;lt;ParameterBinding Name=&quot;dvt_startposition&quot;
Location=&quot;Postback&quot; DefaultValue=&quot;&quot;/&amp;gt;
&amp;lt;ParameterBinding Name=&quot;dvt_firstrow&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
&amp;lt;ParameterBinding Name=&quot;OpenMenuKeyAccessible&quot;
Location=&quot;Resource(wss,OpenMenuKeyAccessible)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;open_menu&quot;
Location=&quot;Resource(wss,open_menu)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;select_deselect_all&quot;
Location=&quot;Resource(wss,select_deselect_all)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;idPresEnabled&quot;
Location=&quot;Resource(wss,idPresEnabled)&quot; /&amp;gt;&amp;lt;ParameterBinding
Name=&quot;NoAnnouncements&quot; Location=&quot;Resource(wss,NoAnnouncements)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;NoAnnouncementsHowTo&quot;
Location=&quot;Resource(wss,NoAnnouncementsHowTo)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;AddNewAnnouncement&quot;
Location=&quot;Resource(wss,idHomePageNewAnnounce)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;MoreAnnouncements&quot;
Location=&quot;Resource(wss,MoreAnnouncements)&quot; /&amp;gt;
&amp;lt;ParameterBinding Name=&quot;ByText&quot;
Location=&quot;Resource(wss,2000)&quot; /&amp;gt;
&lt;/property&gt;
&lt;property name=&quot;Xsl&quot; type=&quot;string&quot; null=&quot;true&quot;
/&gt;
&lt;property name=&quot;CacheXslTimeOut&quot;
type=&quot;int&quot;&gt;86400&lt;/property&gt;
&lt;property name=&quot;WebId&quot; type=&quot;System.Guid, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089&quot;&gt;00000000-0000-
0000-0000-000000000000&lt;/property&gt;
&lt;property name=&quot;ListUrl&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
&lt;property name=&quot;DataSourceID&quot; type=&quot;string&quot; /&gt;
&lt;property name=&quot;FireInitialRow&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
&lt;property name=&quot;ManualRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
&lt;property name=&quot;ViewFlags&quot;
type=&quot;Microsoft.SharePoint.SPViewFlags, Microsoft.SharePoint, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c&quot;&gt;Html, TabularView, Hidden,
Mobile&lt;/property&gt;
&lt;property name=&quot;ChromeState&quot;
type=&quot;chromestate&quot;&gt;Normal&lt;/property&gt;
&lt;property name=&quot;AllowHide&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
&lt;property name=&quot;PageSize&quot; type=&quot;int&quot;&gt;-
1&lt;/property&gt;
&lt;property name=&quot;SampleData&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
&lt;property name=&quot;BaseXsltHashKey&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
&lt;property name=&quot;AsyncRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
```

```
                &lt;property name=&quot;HelpMode&quot;
type=&quot;helpmode&quot;&gt;Modeless&lt;/property&gt;
                &lt;property name=&quot;ListId&quot; type=&quot;System.Guid, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089&quot;&gt;2843ea1b-efe1-
4fcc-a39b-a46121d3efd6&lt;/property&gt;
                &lt;property name=&quot;DataSourceMode&quot;
type=&quot;Microsoft.SharePoint.WebControls.SPDataSourceMode, Microsoft.SharePoint,
Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c&quot;&gt;List&lt;/property&gt;
                &lt;property name=&quot;AllowMinimize&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;TitleUrl&quot;
type=&quot;string&quot;&gt;/Lists/Announcements&lt;/property&gt;
                &lt;property name=&quot;CatalogIconImageUrl&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;DataSourcesString&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;GhostedXslLink&quot;
type=&quot;string&quot;&gt;main.xsl&lt;/property&gt;
                &lt;property name=&quot;PageType&quot; type=&quot;Microsoft.SharePoint.PAGETYPE,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c&quot;&gt;PAGE_NORMALVIEW&lt;/property&gt;
                &lt;property name=&quot;DisplayName&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;UseSQLDataSourcePaging&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;Width&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;ExportMode&quot;
type=&quot;exportmode&quot;&gt;All&lt;/property&gt;
                &lt;property name=&quot;XslLink&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                &lt;property name=&quot;ViewContentTypeId&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;HelpUrl&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;XmlDefinition&quot;
type=&quot;string&quot;&gt;&amp;lt;View Name=&quot;{B438C3E6-2006-404B-9741-
621759847130}&quot; MobileView=&quot;TRUE&quot; Type=&quot;HTML&quot; Hidden=&quot;TRUE&quot;
DisplayName=&quot;&quot; Url=&quot;/default.aspx&quot; Level=&quot;1&quot;
BaseViewID=&quot;0&quot;
ContentTypeID=&quot;0x&quot;&amp;gt;&amp;lt;Query&amp;gt;&amp;lt;Where&amp;gt;&amp;lt;Or&amp;
gt;&amp;lt;IsNull&amp;gt;&amp;lt;FieldRef
Name=&quot;Expires&quot;/&amp;gt;&amp;lt;/IsNull&amp;gt;&amp;lt;Geq&amp;gt;&amp;lt;FieldRef
Name=&quot;Expires&quot;/&amp;gt;&amp;lt;Value
Type=&quot;DateTime&quot;&amp;gt;&amp;lt;Today/&amp;gt;&amp;lt;/Value&amp;gt;&amp;lt;/Geq&amp
;gt;&amp;lt;/Or&amp;gt;&amp;lt;/Where&amp;gt;&amp;lt;OrderBy&amp;gt;&amp;lt;FieldRef
Name=&quot;Modified&quot;
Ascending=&quot;FALSE&quot;/&amp;gt;&amp;lt;/OrderBy&amp;gt;&amp;lt;/Query&amp;gt;&amp;lt;Vie
wFields&amp;gt;&amp;lt;FieldRef Name=&quot;LinkTitleNoMenu&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;&amp;lt;FieldRef Name=&quot;Body&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;&amp;lt;FieldRef Name=&quot;Author&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;&amp;lt;FieldRef Name=&quot;Modified&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;&amp;lt;FieldRef Name=&quot;Attachments&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;&amp;lt;/ViewFields&amp;gt;&amp;lt;RowLimit&amp;gt;5&amp;lt
;/RowLimit&amp;gt;&amp;lt;Toolbar
Type=&quot;Freeform&quot;/&amp;gt;&amp;lt;/View&amp;gt;&lt;/property&gt;
                &lt;property name=&quot;Default&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;TitleIconImageUrl&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;MissingAssembly&quot; type=&quot;string&quot;&gt;Cannot
import this Web Part.&lt;/property&gt;
                &lt;property name=&quot;SelectParameters&quot; type=&quot;string&quot; /&gt;
            &lt;/properties&gt;
            &lt;/data&gt;
            &lt;/webPart&gt;
            &lt;/webParts&gt;</webPartXml>
          <storage>Shared</storage>
          <zoneId></zoneId>
```

```
            <zoneIndex>0</zoneIndex>
         </AddWebPartToZone>
      </soap:Body>
   </soap:Envelope>
```

A sample response is similar to the following:

```
<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>
      <AddWebPartToZoneResponse xmlns="http://microsoft.com/sharepoint/webpartpages">
         <AddWebPartToZoneResult>27b00113-ec28-4ee1-9d35-
bc5bb645a7d0</AddWebPartToZoneResult>
      </AddWebPartToZoneResponse>
   </soap:Body>
</soap:Envelope>
```

4.3.2   Update a Web Part

If the client later needs to change some properties of the Web Part the client requests the properties
of the Web Part again with the value of the **storageKey** that was previously returned.
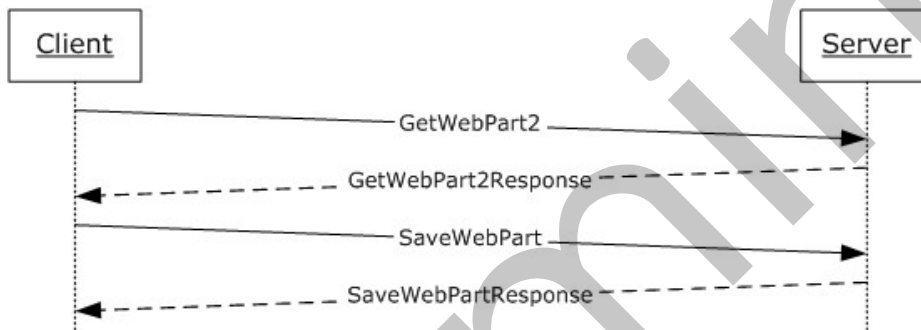
Figure 4: Updating a Web Part

4.3.2.1   Retrieve the Properties of a Web Part

The client can send the **GetWebPart2** request to the server, which includes the recorded
storageKey. The server returns the **GetWebPart2Response** element, which contains the markup
for the Web Part.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <GetWebPart2 xmlns="http://microsoft.com/sharepoint/webpartpages">
         <pageurl>default.aspx</pageurl>
         <storageKey>27b00113-ec28-4ee1-9d35-bc5bb645a7d0</storageKey>
         <storage>Shared</storage>
         <behavior>Version3</behavior>
```

```
        </GetWebPart2>
    </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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>
        <GetWebPart2Response xmlns="http://microsoft.com/sharepoint/webpartpages">
            <GetWebPart2Result>&lt;webParts&gt;
            &lt;webPart xmlns="http://schemas.microsoft.com/WebPart/v3"&gt;
            &lt;metaData&gt;
            &lt;type name="Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"
/&gt;
            &lt;importErrorMessage&gt;Cannot import this Web Part.&lt;/importErrorMessage&gt;
            &lt;/metaData&gt;
            &lt;data&gt;
            &lt;properties&gt;
            &lt;property name="InitialAsyncDataFetch" type="bool"&gt;False&lt;/property&gt;
            &lt;property name="ChromeType" type="chrometype"&gt;Default&lt;/property&gt;
            &lt;property name="Title" type="string" /&gt;
            &lt;property name="Height" type="string" /&gt;
            &lt;property name="CacheXslStorage" type="bool"&gt;True&lt;/property&gt;
            &lt;property name="ListDisplayName" type="string" null="true" /&gt;
            &lt;property name="AllowZoneChange" type="bool"&gt;True&lt;/property&gt;
            &lt;property name="AllowEdit" type="bool"&gt;True&lt;/property&gt;
            &lt;property name="XmlDefinitionLink" type="string" /&gt;
            &lt;property name="DataFields" type="string" /&gt;
            &lt;property name="Hidden" type="bool"&gt;False&lt;/property&gt;
            &lt;property name="ListName" type="string"&gt;{2843EA1B-EFE1-4FCC-A39B-
A46121D3EFD6}&lt;/property&gt;
            &lt;property name="NoDefaultStyle" type="string" null="true" /&gt;
            &lt;property name="AutoRefresh" type="bool"&gt;False&lt;/property&gt;
            &lt;property name="ViewFlag" type="string"&gt;8388745&lt;/property&gt;
            &lt;property name="Direction" type="direction"&gt;NotSet&lt;/property&gt;
            &lt;property name="AutoRefreshInterval" type="int"&gt;60&lt;/property&gt;
            &lt;property name="AllowConnect" type="bool"&gt;True&lt;/property&gt;
            &lt;property name="Description" type="string" /&gt;
            &lt;property name="AllowClose" type="bool"&gt;True&lt;/property&gt;
            &lt;property name="ShowWithSampleData" type="bool"&gt;False&lt;/property&gt;
            &lt;property name="ParameterBindings" type="string"&gt;
            &amp;lt;ParameterBinding Name="dvt_sortdir"
Location="Postback;Connection"/&amp;gt;
            &amp;lt;ParameterBinding Name="dvt_sortfield"
Location="Postback;Connection"/&amp;gt;
            &amp;lt;ParameterBinding Name="dvt_startposition" Location="Postback"
DefaultValue=""/&amp;gt;
            &amp;lt;ParameterBinding Name="dvt_firstrow"
Location="Postback;Connection"/&amp;gt;
            &amp;lt;ParameterBinding Name="OpenMenuKeyAccessible"
Location="Resource(wss,OpenMenuKeyAccessible)" /&amp;gt;
            &amp;lt;ParameterBinding Name="open_menu" Location="Resource(wss,open_menu)"
/&amp;gt;
            &amp;lt;ParameterBinding Name="select_deselect_all"
Location="Resource(wss,select_deselect_all)" /&amp;gt;
```

```
              &amp;lt;ParameterBinding Name="idPresEnabled"
Location="Resource(wss,idPresEnabled)" /&amp;gt;
              &amp;lt;ParameterBinding Name="NoAnnouncements"
Location="Resource(wss,NoAnnouncements)" /&amp;gt;
              &amp;lt;ParameterBinding Name="NoAnnouncementsHowTo"
Location="Resource(wss,NoAnnouncementsHowTo)" /&amp;gt;
              &amp;lt;ParameterBinding Name="AddNewAnnouncement"
Location="Resource(wss,idHomePageNewAnnounce)" /&amp;gt;
              &amp;lt;ParameterBinding Name="MoreAnnouncements"
Location="Resource(wss,MoreAnnouncements)" /&amp;gt;
              &amp;lt;ParameterBinding Name="ByText" Location="Resource(wss,2000)" /&amp;gt;
              &lt;/property&gt;
              &lt;property name="Xsl" type="string" null="true" /&gt;
              &lt;property name="CacheXslTimeOut" type="int"&gt;86400&lt;/property&gt;
              &lt;property name="WebId" type="System.Guid, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"&gt;00000000-0000-0000-0000-
000000000000&lt;/property&gt;
              &lt;property name="ListUrl" type="string" null="true" /&gt;
              &lt;property name="DataSourceID" type="string" /&gt;
              &lt;property name="FireInitialRow" type="bool"&gt;True&lt;/property&gt;
              &lt;property name="ManualRefresh" type="bool"&gt;False&lt;/property&gt;
              &lt;property name="ViewFlags" type="Microsoft.SharePoint.SPViewFlags,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c"&gt;Html, Hidden, FreeForm, Mobile&lt;/property&gt;
              &lt;property name="ChromeState" type="chromestate"&gt;Normal&lt;/property&gt;
              &lt;property name="AllowHide" type="bool"&gt;True&lt;/property&gt;
              &lt;property name="PageSize" type="int"&gt;-1&lt;/property&gt;
              &lt;property name="SampleData" type="string" null="true" /&gt;
              &lt;property name="BaseXsltHashKey" type="string" null="true" /&gt;
              &lt;property name="AsyncRefresh" type="bool"&gt;False&lt;/property&gt;
              &lt;property name="HelpMode" type="helpmode"&gt;Modeless&lt;/property&gt;
              &lt;property name="ListId" type="System.Guid, mscorlib, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"&gt;2843ea1b-efe1-4fcc-a39b-
a46121d3efd6&lt;/property&gt;
              &lt;property name="DataSourceMode"
type="Microsoft.SharePoint.WebControls.SPDataSourceMode, Microsoft.SharePoint,
Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c"&gt;List&lt;/property&gt;
              &lt;property name="AllowMinimize" type="bool"&gt;True&lt;/property&gt;
              &lt;property name="TitleUrl"
type="string"&gt;/Lists/Announcements&lt;/property&gt;
              &lt;property name="CatalogIconImageUrl" type="string" /&gt;
              &lt;property name="DataSourcesString" type="string" /&gt;
              &lt;property name="GhostedXslLink" type="string"&gt;main.xsl&lt;/property&gt;
              &lt;property name="PageType" type="Microsoft.SharePoint.PAGETYPE,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c"&gt;PAGE_NORMALVIEW&lt;/property&gt;
              &lt;property name="DisplayName" type="string" /&gt;
              &lt;property name="UseSQLDataSourcePaging" type="bool"&gt;True&lt;/property&gt;
              &lt;property name="Width" type="string" /&gt;
              &lt;property name="ExportMode" type="exportmode"&gt;All&lt;/property&gt;
              &lt;property name="XslLink" type="string" null="true" /&gt;
              &lt;property name="ViewContentTypeId" type="string" /&gt;
              &lt;property name="HelpUrl" type="string" /&gt;
              &lt;property name="XmlDefinition" type="string"&gt;
              &amp;lt;View Name="{27B00113-EC28-4EE1-9D35-BC5BB645A7D0}" MobileView="TRUE"
Type="HTML" Hidden="TRUE" TabularView="FALSE" FreeForm="TRUE" DisplayName=""
Url="/default.aspx" Level="1" BaseViewID="0" ContentTypeID="0x"&amp;gt;
              &amp;lt;Query&amp;gt;&amp;lt;Where&amp;gt;
              &amp;lt;Or&amp;gt;
              &amp;lt;IsNull&amp;gt;&amp;lt;FieldRef Name="Expires"/&amp;gt;
              &amp;lt;/IsNull&amp;gt;&amp;lt;Geq&amp;gt;
```

```
                &amp;lt;FieldRef Name="Expires"/&amp;gt;
                &amp;lt;Value Type="DateTime"&amp;gt;
                &amp;lt;Today/&amp;gt;
                &amp;lt;/Value&amp;gt;
                &amp;lt;/Geq&amp;gt;
                &amp;lt;/Or&amp;gt;
                &amp;lt;/Where&amp;gt;
                &amp;lt;OrderBy&amp;gt;
                &amp;lt;FieldRef Name="Modified" Ascending="FALSE"/&amp;gt;
                &amp;lt;/OrderBy&amp;gt;
                &amp;lt;/Query&amp;gt;
                &amp;lt;ViewFields&amp;gt;
                &amp;lt;FieldRef Name="LinkTitleNoMenu" Explicit="TRUE"/&amp;gt;
                &amp;lt;FieldRef Name="Body" Explicit="TRUE"/&amp;gt;
                &amp;lt;FieldRef Name="Author" Explicit="TRUE"/&amp;gt;
                &amp;lt;FieldRef Name="Modified" Explicit="TRUE"/&amp;gt;
                &amp;lt;FieldRef Name="Attachments" Explicit="TRUE"/&amp;gt;
                &amp;lt;/ViewFields&amp;gt;
                &amp;lt;RowLimit&amp;gt;5&amp;lt;/RowLimit&amp;gt;
                &amp;lt;Toolbar Type="Freeform"/&amp;gt;
                &amp;lt;/View&amp;gt;&lt;/property&gt;
                &lt;property name="Default" type="string" /&gt;
                &lt;property name="TitleIconImageUrl" type="string" /&gt;
                &lt;property name="MissingAssembly" type="string"&gt;Cannot import this Web
  Part.&lt;/property&gt;
                &lt;property name="SelectParameters" type="string" /&gt;
                &lt;/properties&gt;
                &lt;/data&gt;
                &lt;/webPart&gt;
                &lt;/webParts&gt;
            </GetWebPart2Result>
        </GetWebPart2Response>
    </soap:Body>
</soap:Envelope>
```

4.3.2.2   Update a Web Part

If the client needs to make updates to the Web Part, the client modifies the markup of the Web Part in memory, and then sends the **SaveWebPart** request to the server with the new markup and the recorded **storageKey**. The server commits the changes as requested and responds with the **SaveWebPartResponse** element.

A sample request is similar to the following:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <SaveWebPart2 xmlns="http://microsoft.com/sharepoint/webpartpages">
            <pageUrl>default.aspx</pageUrl>
            <storageKey>27b00113-ec28-4ee1-9d35-bc5bb645a7d0</storageKey>
            <webPartXml>
            &lt;webParts&gt;
            &lt;webPart xmlns=&quot;http://schemas.microsoft.com/WebPart/v3&quot;&gt;
            &lt;metaData&gt;
```

```
                &lt;type name=&quot;Microsoft.SharePoint.WebPartPages.XsltListViewWebPart,
Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c&quot; /&gt;
                &lt;importErrorMessage&gt;Cannot import this Web Part.&lt;/importErrorMessage&gt;
                &lt;/metaData&gt;
                &lt;data&gt;
                &lt;properties&gt;
                &lt;property name=&quot;InitialAsyncDataFetch&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
                &lt;property name=&quot;ChromeType&quot;
type=&quot;chrometype&quot;&gt;Default&lt;/property&gt;
                &lt;property name=&quot;Title&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;Height&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;CacheXslStorage&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;ListDisplayName&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                &lt;property name=&quot;AllowZoneChange&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;AllowEdit&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;XmlDefinitionLink&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;DataFields&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;Hidden&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
                &lt;property name=&quot;ListName&quot; type=&quot;string&quot;&gt;{2843EA1B-EFE1-
4FCC-A39B-A46121D3EFD6}&lt;/property&gt;
                &lt;property name=&quot;NoDefaultStyle&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                &lt;property name=&quot;AutoRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
                &lt;property name=&quot;ViewFlag&quot;
type=&quot;string&quot;&gt;8388745&lt;/property&gt;
                &lt;property name=&quot;Direction&quot;
type=&quot;direction&quot;&gt;NotSet&lt;/property&gt;
                &lt;property name=&quot;AutoRefreshInterval&quot;
type=&quot;int&quot;&gt;60&lt;/property&gt;
                &lt;property name=&quot;AllowConnect&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;Description&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;AllowClose&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                &lt;property name=&quot;ShowWithSampleData&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
                &lt;property name=&quot;ParameterBindings&quot; type=&quot;string&quot;&gt;
&amp;lt;ParameterBinding Name=&quot;dvt_sortdir&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;dvt_sortfield&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;dvt_startposition&quot;
Location=&quot;Postback&quot; DefaultValue=&quot;&quot;/&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;dvt_firstrow&quot;
Location=&quot;Postback;Connection&quot;/&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;OpenMenuKeyAccessible&quot;
Location=&quot;Resource(wss,OpenMenuKeyAccessible)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;open_menu&quot;
Location=&quot;Resource(wss,open_menu)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;select_deselect_all&quot;
Location=&quot;Resource(wss,select_deselect_all)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;idPresEnabled&quot;
Location=&quot;Resource(wss,idPresEnabled)&quot; /&amp;gt;&amp;lt;ParameterBinding
Name=&quot;NoAnnouncements&quot; Location=&quot;Resource(wss,NoAnnouncements)&quot; /&amp;gt;
```

```
                &amp;lt;ParameterBinding Name=&quot;NoAnnouncementsHowTo&quot;
Location=&quot;Resource(wss,NoAnnouncementsHowTo)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;AddNewAnnouncement&quot;
Location=&quot;Resource(wss,idHomePageNewAnnounce)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;MoreAnnouncements&quot;
Location=&quot;Resource(wss,MoreAnnouncements)&quot; /&amp;gt;
                &amp;lt;ParameterBinding Name=&quot;ByText&quot;
Location=&quot;Resource(wss,2000)&quot; /&amp;gt;&lt;/property&gt;
            &lt;property name=&quot;Xsl&quot; type=&quot;string&quot; null=&quot;true&quot;
/&gt;
            &lt;property name=&quot;CacheXslTimeOut&quot;
type=&quot;int&quot;&gt;86400&lt;/property&gt;
            &lt;property name=&quot;WebId&quot; type=&quot;System.Guid, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089&quot;&gt;00000000-0000-
0000-0000-000000000000&lt;/property&gt;
            &lt;property name=&quot;ListUrl&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
            &lt;property name=&quot;DataSourceID&quot; type=&quot;string&quot; /&gt;
            &lt;property name=&quot;FireInitialRow&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
            &lt;property name=&quot;ManualRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
            &lt;property name=&quot;ViewFlags&quot;
type=&quot;Microsoft.SharePoint.SPViewFlags, Microsoft.SharePoint, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c&quot;&gt;Html, Hidden, FreeForm,
Mobile&lt;/property&gt;
                    &lt;property name=&quot;ChromeState&quot;
type=&quot;chromestate&quot;&gt;Normal&lt;/property&gt;
                    &lt;property name=&quot;AllowHide&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                    &lt;property name=&quot;PageSize&quot; type=&quot;int&quot; &gt;-
1&lt;/property&gt;
                    &lt;property name=&quot;SampleData&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                    &lt;property name=&quot;BaseXsltHashKey&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                    &lt;property name=&quot;AsyncRefresh&quot;
type=&quot;bool&quot;&gt;False&lt;/property&gt;
                    &lt;property name=&quot;HelpMode&quot;
type=&quot;helpmode&quot;&gt;Modeless&lt;/property&gt;
                    &lt;property name=&quot;ListId&quot; type=&quot;System.Guid, mscorlib,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089&quot;&gt;2843ea1b-efe1-
4fcc-a39b-a46121d3efd6&lt;/property&gt;
                    &lt;property name=&quot;DataSourceMode&quot;
type=&quot;Microsoft.SharePoint.WebControls.SPDataSourceMode, Microsoft.SharePoint,
Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c&quot;&gt;List&lt;/property&gt;
                    &lt;property name=&quot;AllowMinimize&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                    &lt;property name=&quot;TitleUrl&quot;
type=&quot;string&quot;&gt;/Lists/Announcements&lt;/property&gt;
                    &lt;property name=&quot;CatalogIconImageUrl&quot; type=&quot;string&quot;
/&gt;
                    &lt;property name=&quot;DataSourcesString&quot; type=&quot;string&quot;
/&gt;
                    &lt;property name=&quot;GhostedXslLink&quot;
type=&quot;string&quot;&gt;main.xsl&lt;/property&gt;
                    &lt;property name=&quot;PageType&quot;
type=&quot;Microsoft.SharePoint.PAGETYPE, Microsoft.SharePoint, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=71e9bce111e9429c&quot;&gt;PAGE_NORMALVIEW&lt;/property&gt;
                    &lt;property name=&quot;DisplayName&quot; type=&quot;string&quot; /&gt;
```

```
                    &lt;property name=&quot;UseSQLDataSourcePaging&quot;
type=&quot;bool&quot;&gt;True&lt;/property&gt;
                    &lt;property name=&quot;Width&quot; type=&quot;string&quot;
&gt;88&lt;/property&gt;
                    &lt;property name=&quot;ExportMode&quot;
type=&quot;exportmode&quot;&gt;All&lt;/property&gt;
                    &lt;property name=&quot;XslLink&quot; type=&quot;string&quot;
null=&quot;true&quot; /&gt;
                    &lt;property name=&quot;ViewContentTypeId&quot; type=&quot;string&quot;
/&gt;
                    &lt;property name=&quot;HelpUrl&quot; type=&quot;string&quot; /&gt;
                    &lt;property name=&quot;XmlDefinition&quot;
type=&quot;string&quot;&gt;&amp;lt;View Name=&quot;{27B00113-EC28-4EE1-9D35-
BC5BB645A7D0}&quot; MobileView=&quot;TRUE&quot; Type=&quot;HTML&quot; Hidden=&quot;TRUE&quot;
TabularView=&quot;FALSE&quot; FreeForm=&quot;TRUE&quot; DisplayName=&quot;&quot;
Url=&quot;/default.aspx&quot; Level=&quot;1&quot; BaseViewID=&quot;0&quot;
ContentTypeID=&quot;0x&quot;&amp;gt;
                    &amp;lt;Query&amp;gt;
                    &amp;lt;Where&amp;gt;
                    &amp;lt;Or&amp;gt;
                    &amp;lt;IsNull&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Expires&quot;/&amp;gt;
                    &amp;lt;/IsNull&amp;gt;
                    &amp;lt;Geq&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Expires&quot;/&amp;gt;
                    &amp;lt;Value Type=&quot;DateTime&quot;
&amp;gt;&amp;lt;Today/&amp;gt;
                    &amp;lt;/Value&amp;gt;&amp;lt;/Geq&amp;gt;
                    &amp;lt;/Or&amp;gt;&amp;lt;/Where&amp;gt;
                    &amp;lt;OrderBy&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Modified&quot;
Ascending=&quot;FALSE&quot;/&amp;gt;
                    &amp;lt;/OrderBy&amp;gt;&amp;lt;/Query&amp;gt;
                    &amp;lt;ViewFields&amp;gt;
                    &amp;lt;FieldRef Name=&quot;LinkTitleNoMenu&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Body&quot; Explicit=&quot;TRUE&quot;/&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Author&quot; Explicit=&quot;TRUE&quot;/&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Modified&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;
                    &amp;lt;FieldRef Name=&quot;Attachments&quot;
Explicit=&quot;TRUE&quot;/&amp;gt;

&amp;lt;/ViewFields&amp;gt;&amp;lt;RowLimit&amp;gt;5&amp;lt;/RowLimit&amp;gt;
                    &amp;lt;Toolbar Type=&quot;Freeform&quot;/&amp;gt;
                    &amp;lt;/View&amp;gt;
                &lt;/property&gt;
                &lt;property name=&quot;Default&quot; type=&quot;string&quot; /&gt;
                &lt;property name=&quot;TitleIconImageUrl&quot; type=&quot;string&quot;
/&gt;
                &lt;property name=&quot;MissingAssembly&quot;
type=&quot;string&quot;&gt;Cannot import this Web Part.&lt;/property&gt;
                &lt;property name=&quot;SelectParameters&quot; type=&quot;string&quot;
/&gt;
                &lt;/properties&gt;
                &lt;/data&gt;
                &lt;/webPart&gt;
                &lt;/webParts&gt;
            </webPartXml>
          <storage>Shared</storage>
       <allowTypeChange>false</allowTypeChange>
```

```
        </SaveWebPart2>
    </soap:Body>
</soap:Envelope>
```

A sample response is similar to the following:

```
<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><SaveWebPart2Response
xmlns="http://microsoft.com/sharepoint/webpartpages" /></soap:Body></soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

In general, these operations are meant to be called by users with permissions to create and edit pages on the site (2), not by users who are simply browsing the site (2). In certain cases, the server could require higher privileges. Even innocuous operations contain the possibility of exposing unintended details about the implementation and operation of the site (2). Take care to ensure that only the proper set of users can call these operations.

5.2 Index of Security Parameters

None.

6   Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided int his appendix.

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://microsoft.com/sharepoint/webpartpages"
xmlns:s1="http://microsoft.com/wsdl/types/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://microsoft.com/sharepoint/webpartpages"
xmlns:wpv2="http://schemas.microsoft.com/WebPart/v2"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/sharepoint/webpartpages">
        <s:import namespace="http://schemas.microsoft.com/WebPart/v2" />
        <s:import namespace="http://microsoft.com/wsdl/types/" />
        <s:import namespace="http://www.w3.org/2001/XMLSchema" />
        <s:element name="GetWebPartPageDocument">
   <s:complexType>
     <s:sequence>
       <s:element name="documentName" type="s:string" minOccurs="1" maxOccurs="1"/>
     </s:sequence>
   </s:complexType>
</s:element>
        <s:element name="GetWebPartPageDocumentResponse">
   <s:complexType>
     <s:sequence>
       <s:element name="GetWebPartPageDocumentResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
     </s:sequence>
   </s:complexType>
</s:element>
<s:element name="GetWebPartPage">
   <s:complexType>
     <s:sequence>
       <s:element name="documentName" type="s:string" minOccurs="1" maxOccurs="1"/>
       <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1"/>
     </s:sequence>
   </s:complexType>
</s:element>
        <s:simpleType name="SPWebServiceBehavior">
   <s:restriction base="s:string">
     <s:enumeration value="Version3"/>
   </s:restriction>
</s:simpleType>
<s:element name="GetWebPartPageResponse">
   <s:complexType>
     <s:sequence>
       <s:element name="GetWebPartPageResult" type="s:string" minOccurs="1" maxOccurs="1"/>
     </s:sequence>
   </s:complexType>
</s:element>
<s:element name="RenderWebPartForEdit">
   <s:complexType>
     <s:sequence>
       <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
```

```
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="RenderWebPartForEditResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="RenderWebPartForEditResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetXmlDataFromDataSource">
        <s:complexType>
          <s:sequence>
            <s:element name="queryXml" type="s:string" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetXmlDataFromDataSourceResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetXmlDataFromDataSourceResult" type="s:string" minOccurs="1"
maxOccurs="1" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFormCapabilityFromDataSourceControl">
        <s:complexType>
          <s:sequence>
            <s:element name="dscXml" type="s:string" minOccurs="1" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetFormCapabilityFromDataSourceControlResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetFormCapabilityFromDataSourceControlResult" type="s:string"
minOccurs="1" maxOccurs="1" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetDataFromDataSourceControl">
        <s:complexType>
          <s:sequence>
            <s:element name="dscXml" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="contextUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetDataFromDataSourceControlResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetDataFromDataSourceControlResult" type="s:string" minOccurs="1"
maxOccurs="1" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetExpandedListViewXml">
        <s:complexType>
```

```
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="webId" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="listViewXml" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="listGuid" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="listUrl" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="listDisplayName" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="hasConnection" type="s:boolean" />
      <s:element minOccurs="0" maxOccurs="1" name="partialView" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:element name="GetExpandedListViewXmlResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="GetExpandedListViewXmlResult"
type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>

      <s:element name="FetchLegalWorkflowActions">
  <s:complexType/>
</s:element>
      <s:element name="FetchLegalWorkflowActionsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="FetchLegalWorkflowActionsResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
      <s:element name="ValidateWorkflowMarkupAndCreateSupportObjects">
  <s:complexType>
    <s:sequence>
      <s:element name="workflowMarkupText" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="rulesText" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="configBlob" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="flag" type="s:int" minOccurs="1" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
      <s:element name="ValidateWorkflowMarkupAndCreateSupportObjectsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="ValidateWorkflowMarkupAndCreateSupportObjectsResult"
type="s:string" minOccurs="1" maxOccurs="1" />
    </s:sequence>
  </s:complexType>
</s:element>
      <s:element name="AssociateWorkflowMarkup">
  <s:complexType>
    <s:sequence>
      <s:element name="configUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
      <s:element name="configVersion" minOccurs="0" maxOccurs="1">
        <s:simpleType>
          <s:restriction base="s:string">
            <s:pattern value="[v|V]\d+\.\d+" />
          </s:restriction>
        </s:simpleType>
```

171 / 224

```
            </s:element>

        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="AssociateWorkflowMarkupResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="AssociateWorkflowMarkupResult" type="s:string" minOccurs="1"
maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="RemoveWorkflowAssociation">
    <s:complexType>
        <s:sequence>
            <s:element name="configUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="configVersion" minOccurs="0" maxOccurs="1">
                <s:simpleType>
                    <s:restriction base="s:string">
                        <s:pattern value="[v|V]\d+\.\d+" />
                    </s:restriction>
                </s:simpleType>
            </s:element>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="RemoveWorkflowAssociationResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="RemoveWorkflowAssociationResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="ConvertWebPartFormat">
    <s:complexType>
        <s:sequence>
            <s:element name="inputFormat" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="formatConversionOption" type="tns:FormatConversionOption"
minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:simpleType name="FormatConversionOption">
    <s:restriction base="s:string">
        <s:enumeration value="ConvertToWebPartExportFormat"/>
        <s:enumeration value="ConvertToWebPartDesignerPersistenceFormat"/>
    </s:restriction>
</s:simpleType>
        <s:element name="ConvertWebPartFormatResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="ConvertWebPartFormatResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetAssemblyMetaData">
```

```
        <s:complexType>
            <s:sequence>
                <s:element name="assemblyName" type="s:string" minOccurs="1" maxOccurs="1"/>
                <s:element name="baseTypes" minOccurs="1" maxOccurs="1">
                    <s:simpleType>
                        <s:restriction base="s:string">
                            <s:pattern value="([^\n\r;]+;)*[^\n\r;]+ " />
                        </s:restriction>
                    </s:simpleType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
        <s:element name="GetAssemblyMetaDataResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetAssemblyMetaDataResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetBindingResourceData">
    <s:complexType>
        <s:sequence>
            <s:element name="ResourceName" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetBindingResourceDataResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetBindingResourceDataResult" type="s:string" minOccurs="1"
maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="ExecuteProxyUpdates">
    <s:complexType>
        <s:sequence>
            <s:element name="updateData" type="s:string" minOccurs="0" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="ExecuteProxyUpdatesResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="ExecuteProxyUpdatesResult" type="s:string" minOccurs="1"
maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="AddWebPartToZone">
    <s:complexType>
        <s:sequence>
            <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="storage" type="tns:Storage" minOccurs="1" maxOccurs="1"/>
            <s:element name="zoneId" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="zoneIndex" type="s:int" minOccurs="0" maxOccurs="1"/>
```

```
        </s:sequence>
      </s:complexType>
</s:element>
          <s:simpleType name="Storage">
    <s:restriction base="s:string">
       <s:enumeration value="None"/>
       <s:enumeration value="Personal"/>
       <s:enumeration value="Shared"/>
    </s:restriction>
</s:simpleType>
          <s:element name="AddWebPartToZoneResponse">
    <s:complexType>
       <s:sequence>
          <s:element name="AddWebPartToZoneResult" type="s1:guid" minOccurs="1"
maxOccurs="1"/>
       </s:sequence>
    </s:complexType>
</s:element>
          <s:element name="AddWebPart">
    <s:complexType>
       <s:sequence>
          <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="storage" type="tns:Storage" minOccurs="1" maxOccurs="1"/>
       </s:sequence>
    </s:complexType>
</s:element>
          <s:element name="AddWebPartResponse">
    <s:complexType>
       <s:sequence>
          <s:element name="AddWebPartResult" type="s1:guid" minOccurs="1" maxOccurs="1" />
       </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetWebPart">
    <s:complexType>
       <s:sequence>
          <s:element name="pageurl" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
          <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
       </s:sequence>
    </s:complexType>
</s:element>
          <s:element name="GetWebPartResponse">
    <s:complexType>
       <s:sequence>
          <s:element name="GetWebPartResult" type="s:string" minOccurs="1" maxOccurs="1" />
       </s:sequence>
    </s:complexType>
</s:element>
          <s:element name="GetWebPart2">
    <s:complexType>
       <s:sequence>
          <s:element name="pageurl" type="s:string" minOccurs="1" maxOccurs="1"/>
          <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
          <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
          <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1" />
       </s:sequence>
```

```
            </s:complexType>
    </s:element>
            <s:element name="GetWebPart2Response">
    <s:complexType>
        <s:sequence>
            <s:element name="GetWebPart2Result" type="s:string" minOccurs="1" maxOccurs="1" />
        </s:sequence>
    </s:complexType>
    </s:element>
            <s:element name="GetCustomControlList">
    <s:complexType/>
    </s:element>
            <s:element name="GetCustomControlListResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetCustomControlListResult" minOccurs="1" maxOccurs="1">
                <s:complexType>
                    <s:sequence>
                        <s:element name="ControlsList" minOccurs="1" maxOccurs="1">
                            <s:complexType>
                                <s:sequence>
                                    <s:element name="Assembly" minOccurs="0" maxOccurs="unbounded">
                                        <s:complexType>
                                            <s:sequence>
                                                <s:element name="Control" minOccurs="1"
maxOccurs="unbounded">
                                                    <s:complexType>
                                                        <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>
                                                </s:element>
                                            </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="optional"/>
                                        </s:complexType>
                                    </s:element>
                                </s:sequence>
                            </s:complexType>
                        </s:element>
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:sequence>
    </s:complexType>
    </s:element>
            <s:element name="GetSafeAssemblyInfo">
    <s:complexType/>
    </s:element>
            <s:element name="GetSafeAssemblyInfoResponse">
    <s:complexType>
```

```
<s:sequence>
    <s:element name="GetSafeAssemblyInfoResult" minOccurs="1" maxOccurs="1">
        <s:complexType>
            <s:sequence>
                <s:element name="SafeAssemblyInfo" minOccurs="1" maxOccurs="1">
                    <s:complexType mixed="true">
                        <s:sequence>
                            <s:element name="Assembly" minOccurs="0" maxOccurs="unbounded" >
                                <s:complexType>
                                    <s:sequence />
                                    <s:attribute name="Name" type="s:string" use="required"/>
                                    <s:attribute name="FullName" type="s:string"
use="required"/>
                                </s:complexType>
                            </s:element>
                        </s:sequence>
                    </s:complexType>
                </s:element>
            </s:sequence>
        </s:complexType>
    </s:element>
</s:sequence>
</s:complexType>
</s:element>
        <s:element name="GetWebPartProperties">
    <s:complexType>
        <s:sequence>
            <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
<s:element name="GetWebPartPropertiesResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetWebPartPropertiesResult" minOccurs="1" maxOccurs="1">
                <s:complexType mixed="true">
                    <s:sequence>
                        <s:element name="WebParts" minOccurs="1" maxOccurs="1">
                            <s:complexType mixed="true">
                                <s:sequence>
                                    <s:element name="WebPart" type="wpv2:WebPart" minOccurs="0"
maxOccurs="unbounded" />
                                </s:sequence>
                            </s:complexType>
                        </s:element>
                    </s:sequence>
                </s:complexType>
            </s:element>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetWebPartProperties2">
    <s:complexType>
        <s:sequence>
            <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
            <s:element name="behavior" type="tns:SPWebServiceBehavior" minOccurs="0"
maxOccurs="1"/>
```

```xml
      </s:sequence>
   </s:complexType>
</s:element>
         <s:element name="GetWebPartProperties2Response">
   <s:complexType>
      <s:sequence>
         <s:element name="GetWebPartProperties2Result" minOccurs="1" maxOccurs="1">
             <s:complexType mixed="true">
                <s:sequence>
                   <s:element name="WebParts" minOccurs="1" maxOccurs="1">
                       <s:complexType mixed="true">
                          <s:sequence>
                             <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
                          </s:sequence>
                       </s:complexType>
                   </s:element>
                </s:sequence>
             </s:complexType>
         </s:element>
      </s:sequence>
   </s:complexType>
</s:element>
         <s:element name="SaveWebPart">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
         <s:element name="SaveWebPartResponse">
   <s:complexType/>
</s:element>
         <s:element name="SaveWebPart2">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1" />
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1" />
         <s:element name="webPartXml" type="s:string" minOccurs="1" maxOccurs="1" />
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1" />
         <s:element name="allowTypeChange" type="s:boolean" minOccurs="0"  maxOccurs="1" />
      </s:sequence>
   </s:complexType>
</s:element>
         <s:element name="SaveWebPart2Response">
   <s:complexType/>
</s:element>
         <s:element name="DeleteWebPart">
   <s:complexType>
      <s:sequence>
         <s:element name="pageUrl" type="s:string" minOccurs="1" maxOccurs="1"/>
         <s:element name="storageKey" type="s1:guid" minOccurs="1" maxOccurs="1"/>
         <s:element name="storage" type="tns:Storage" minOccurs="0" maxOccurs="1"/>
      </s:sequence>
   </s:complexType>
</s:element>
         <s:element name="DeleteWebPartResponse">
```

```
        <s:complexType/>
</s:element>
        <s:element name="GetWebPartPageConnectionInfo">
    <s:complexType>
        <s:sequence>
            <s:element name="sourcePageUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="sourcePageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="providerPartID" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="lcid" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetWebPartPageConnectionInfoResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetWebPartPageConnectionInfoResult" type="s:string" minOccurs="1"
maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetWebPartCrossPageCompatibility">
    <s:complexType>
        <s:sequence>
            <s:element name="sourcePageUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="sourcePageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="targetPageUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="targetPageContents" type="s:string" minOccurs="0" maxOccurs="1"/>
            <s:element name="providerPartID" type="s:string" minOccurs="1" maxOccurs="1"/>
            <s:element name="lcid" type="s:string" minOccurs="1" maxOccurs="1"/>
        </s:sequence>
    </s:complexType>
</s:element>
        <s:element name="GetWebPartCrossPageCompatibilityResponse">
    <s:complexType>
        <s:sequence>
            <s:element name="GetWebPartCrossPageCompatibilityResult" type="s:string"
minOccurs="1" maxOccurs="1" />
        </s:sequence>
    </s:complexType>
</s:element>
        </s:schema>
        <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wsdl/types/">
        <s:simpleType name="guid">
            <s:restriction base="s:string">
                <s:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{12}" />
            </s:restriction>
        </s:simpleType>
        </s:schema>
        <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/WebPart/v2">
        <s:complexType name="WebPart">
            <s:sequence minOccurs="2" maxOccurs="unbounded">
                <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
            </s:sequence>
            <s:attribute name="ID" type="s:string" use="optional" />
            <s:anyAttribute namespace="##any"/>
        </s:complexType>
```

```
            </s:schema>
            <s:schema xmlns:s="http://www.w3.org/2001/XMLSchema" targetNamespace="
      http://schemas.microsoft.com/sharepoint/soap">
                <s:complexType name="SOAPFaultDetails">
                    <s:sequence>
                        <s:element name="errorstring" type="s:string"/>
                        <s:element name="errorcode" type="s:string" minOccurs="0"/>
                    </s:sequence>
                </s:complexType>
            </s:schema>
      </wsdl:types>
      <wsdl:message name="GetWebPartPageDocumentSoapIn">
          <wsdl:part name="parameters" element="tns:GetWebPartPageDocument" />
      </wsdl:message>
      <wsdl:message name="GetWebPartPageDocumentSoapOut">
          <wsdl:part name="parameters" element="tns:GetWebPartPageDocumentResponse" />
      </wsdl:message>
      <wsdl:message name="GetWebPartPageSoapIn">
          <wsdl:part name="parameters" element="tns:GetWebPartPage" />
      </wsdl:message>
      <wsdl:message name="GetWebPartPageSoapOut">
          <wsdl:part name="parameters" element="tns:GetWebPartPageResponse" />
      </wsdl:message>
      <wsdl:message name="RenderWebPartForEditSoapIn">
          <wsdl:part name="parameters" element="tns:RenderWebPartForEdit" />
      </wsdl:message>
      <wsdl:message name="RenderWebPartForEditSoapOut">
          <wsdl:part name="parameters" element="tns:RenderWebPartForEditResponse" />
      </wsdl:message>
      <wsdl:message name="GetXmlDataFromDataSourceSoapIn">
          <wsdl:part name="parameters" element="tns:GetXmlDataFromDataSource" />
      </wsdl:message>
      <wsdl:message name="GetXmlDataFromDataSourceSoapOut">
          <wsdl:part name="parameters" element="tns:GetXmlDataFromDataSourceResponse" />
      </wsdl:message>
      <wsdl:message name="GetFormCapabilityFromDataSourceControlSoapIn">
          <wsdl:part name="parameters" element="tns:GetFormCapabilityFromDataSourceControl" />
      </wsdl:message>
      <wsdl:message name="GetFormCapabilityFromDataSourceControlSoapOut">
          <wsdl:part name="parameters"
      element="tns:GetFormCapabilityFromDataSourceControlResponse" />
      </wsdl:message>
      <wsdl:message name="GetDataFromDataSourceControlSoapIn">
          <wsdl:part name="parameters" element="tns:GetDataFromDataSourceControl" />
      </wsdl:message>
      <wsdl:message name="GetDataFromDataSourceControlSoapOut">
          <wsdl:part name="parameters" element="tns:GetDataFromDataSourceControlResponse" />
      </wsdl:message>
      <wsdl:message name="GetExpandedListViewXmlSoapIn">
        <wsdl:part name="parameters" element="tns:GetExpandedListViewXml" />
      </wsdl:message>
      <wsdl:message name="GetExpandedListViewXmlSoapOut">
        <wsdl:part name="parameters" element="tns:GetExpandedListViewXmlResponse" />
      </wsdl:message>
      <wsdl:message name="FetchLegalWorkflowActionsSoapIn">
          <wsdl:part name="parameters" element="tns:FetchLegalWorkflowActions" />
      </wsdl:message>
      <wsdl:message name="FetchLegalWorkflowActionsSoapOut">
          <wsdl:part name="parameters" element="tns:FetchLegalWorkflowActionsResponse" />
```

```
      </wsdl:message>
      <wsdl:message name="ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn">
         <wsdl:part name="parameters"
  element="tns:ValidateWorkflowMarkupAndCreateSupportObjects" />
      </wsdl:message>
      <wsdl:message name="ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut">
         <wsdl:part name="parameters"
  element="tns:ValidateWorkflowMarkupAndCreateSupportObjectsResponse" />
      </wsdl:message>
      <wsdl:message name="AssociateWorkflowMarkupSoapIn">
         <wsdl:part name="parameters" element="tns:AssociateWorkflowMarkup" />
      </wsdl:message>
      <wsdl:message name="AssociateWorkflowMarkupSoapOut">
         <wsdl:part name="parameters" element="tns:AssociateWorkflowMarkupResponse" />
      </wsdl:message>
      <wsdl:message name="RemoveWorkflowAssociationSoapIn">
         <wsdl:part name="parameters" element="tns:RemoveWorkflowAssociation" />
      </wsdl:message>
      <wsdl:message name="RemoveWorkflowAssociationSoapOut">
         <wsdl:part name="parameters" element="tns:RemoveWorkflowAssociationResponse" />
      </wsdl:message>
      <wsdl:message name="ConvertWebPartFormatSoapIn">
         <wsdl:part name="parameters" element="tns:ConvertWebPartFormat" />
      </wsdl:message>
      <wsdl:message name="ConvertWebPartFormatSoapOut">
         <wsdl:part name="parameters" element="tns:ConvertWebPartFormatResponse" />
      </wsdl:message>
      <wsdl:message name="GetAssemblyMetaDataSoapIn">
         <wsdl:part name="parameters" element="tns:GetAssemblyMetaData" />
      </wsdl:message>
      <wsdl:message name="GetAssemblyMetaDataSoapOut">
         <wsdl:part name="parameters" element="tns:GetAssemblyMetaDataResponse" />
      </wsdl:message>
      <wsdl:message name="GetBindingResourceDataSoapIn">
         <wsdl:part name="parameters" element="tns:GetBindingResourceData" />
      </wsdl:message>
      <wsdl:message name="GetBindingResourceDataSoapOut">
         <wsdl:part name="parameters" element="tns:GetBindingResourceDataResponse" />
      </wsdl:message>
      <wsdl:message name="ExecuteProxyUpdatesSoapIn">
         <wsdl:part name="parameters" element="tns:ExecuteProxyUpdates" />
      </wsdl:message>
      <wsdl:message name="ExecuteProxyUpdatesSoapOut">
         <wsdl:part name="parameters" element="tns:ExecuteProxyUpdatesResponse" />
      </wsdl:message>
      <wsdl:message name="AddWebPartToZoneSoapIn">
         <wsdl:part name="parameters" element="tns:AddWebPartToZone" />
      </wsdl:message>
      <wsdl:message name="AddWebPartToZoneSoapOut">
         <wsdl:part name="parameters" element="tns:AddWebPartToZoneResponse" />
      </wsdl:message>
      <wsdl:message name="AddWebPartSoapIn">
         <wsdl:part name="parameters" element="tns:AddWebPart" />
      </wsdl:message>
      <wsdl:message name="AddWebPartSoapOut">
         <wsdl:part name="parameters" element="tns:AddWebPartResponse" />
      </wsdl:message>
      <wsdl:message name="GetWebPartSoapIn">
         <wsdl:part name="parameters" element="tns:GetWebPart" />
```

```
    </wsdl:message>
    <wsdl:message name="GetWebPartSoapOut">
       <wsdl:part name="parameters" element="tns:GetWebPartResponse" />
    </wsdl:message>
    <wsdl:message name="GetWebPart2SoapIn">
       <wsdl:part name="parameters" element="tns:GetWebPart2" />
    </wsdl:message>
    <wsdl:message name="GetWebPart2SoapOut">
       <wsdl:part name="parameters" element="tns:GetWebPart2Response" />
    </wsdl:message>
    <wsdl:message name="GetCustomControlListSoapIn">
       <wsdl:part name="parameters" element="tns:GetCustomControlList" />
    </wsdl:message>
    <wsdl:message name="GetCustomControlListSoapOut">
       <wsdl:part name="parameters" element="tns:GetCustomControlListResponse" />
    </wsdl:message>
    <wsdl:message name="GetSafeAssemblyInfoSoapIn">
       <wsdl:part name="parameters" element="tns:GetSafeAssemblyInfo" />
    </wsdl:message>
    <wsdl:message name="GetSafeAssemblyInfoSoapOut">
       <wsdl:part name="parameters" element="tns:GetSafeAssemblyInfoResponse" />
    </wsdl:message>
    <wsdl:message name="GetWebPartPropertiesSoapIn">
       <wsdl:part name="parameters" element="tns:GetWebPartProperties" />
    </wsdl:message>
    <wsdl:message name="GetWebPartPropertiesSoapOut">
       <wsdl:part name="parameters" element="tns:GetWebPartPropertiesResponse" />
    </wsdl:message>
    <wsdl:message name="GetWebPartProperties2SoapIn">
       <wsdl:part name="parameters" element="tns:GetWebPartProperties2" />
    </wsdl:message>
    <wsdl:message name="GetWebPartProperties2SoapOut">
       <wsdl:part name="parameters" element="tns:GetWebPartProperties2Response" />
    </wsdl:message>
    <wsdl:message name="SaveWebPartSoapIn">
       <wsdl:part name="parameters" element="tns:SaveWebPart" />
    </wsdl:message>
    <wsdl:message name="SaveWebPartSoapOut">
       <wsdl:part name="parameters" element="tns:SaveWebPartResponse" />
    </wsdl:message>
    <wsdl:message name="SaveWebPart2SoapIn">
       <wsdl:part name="parameters" element="tns:SaveWebPart2" />
    </wsdl:message>
    <wsdl:message name="SaveWebPart2SoapOut">
       <wsdl:part name="parameters" element="tns:SaveWebPart2Response" />
    </wsdl:message>
    <wsdl:message name="DeleteWebPartSoapIn">
       <wsdl:part name="parameters" element="tns:DeleteWebPart" />
    </wsdl:message>
    <wsdl:message name="DeleteWebPartSoapOut">
       <wsdl:part name="parameters" element="tns:DeleteWebPartResponse" />
    </wsdl:message>
    <wsdl:message name="GetWebPartPageConnectionInfoSoapIn">
       <wsdl:part name="parameters" element="tns:GetWebPartPageConnectionInfo" />
    </wsdl:message>
    <wsdl:message name="GetWebPartPageConnectionInfoSoapOut">
       <wsdl:part name="parameters" element="tns:GetWebPartPageConnectionInfoResponse" />
    </wsdl:message>
    <wsdl:message name="GetWebPartCrossPageCompatibilitySoapIn">
```

```
            <wsdl:part name="parameters" element="tns:GetWebPartCrossPageCompatibility" />
         </wsdl:message>
         <wsdl:message name="GetWebPartCrossPageCompatibilitySoapOut">
            <wsdl:part name="parameters" element="tns:GetWebPartCrossPageCompatibilityResponse" />
         </wsdl:message>
         <wsdl:portType name="WebPartPagesWebServiceSoap">
            <wsdl:operation name="GetWebPartPageDocument">
               <wsdl:input message="tns:GetWebPartPageDocumentSoapIn" />
               <wsdl:output message="tns:GetWebPartPageDocumentSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetWebPartPage">
               <wsdl:input message="tns:GetWebPartPageSoapIn" />
               <wsdl:output message="tns:GetWebPartPageSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="RenderWebPartForEdit">
               <wsdl:input message="tns:RenderWebPartForEditSoapIn" />
               <wsdl:output message="tns:RenderWebPartForEditSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetXmlDataFromDataSource">
               <wsdl:input message="tns:GetXmlDataFromDataSourceSoapIn" />
               <wsdl:output message="tns:GetXmlDataFromDataSourceSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetFormCapabilityFromDataSourceControl">
               <wsdl:input message="tns:GetFormCapabilityFromDataSourceControlSoapIn" />
               <wsdl:output message="tns:GetFormCapabilityFromDataSourceControlSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetDataFromDataSourceControl">
               <wsdl:input message="tns:GetDataFromDataSourceControlSoapIn" />
               <wsdl:output message="tns:GetDataFromDataSourceControlSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetExpandedListViewXml">
               <wsdl:input message="tns:GetExpandedListViewXmlSoapIn" />
               <wsdl:output message="tns:GetExpandedListViewXmlSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="FetchLegalWorkflowActions">
               <wsdl:input message="tns:FetchLegalWorkflowActionsSoapIn" />
               <wsdl:output message="tns:FetchLegalWorkflowActionsSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="ValidateWorkflowMarkupAndCreateSupportObjects">
               <wsdl:input message="tns:ValidateWorkflowMarkupAndCreateSupportObjectsSoapIn" />
               <wsdl:output message="tns:ValidateWorkflowMarkupAndCreateSupportObjectsSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="AssociateWorkflowMarkup">
               <wsdl:input message="tns:AssociateWorkflowMarkupSoapIn" />
               <wsdl:output message="tns:AssociateWorkflowMarkupSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="RemoveWorkflowAssociation">
               <wsdl:input message="tns:RemoveWorkflowAssociationSoapIn" />
               <wsdl:output message="tns:RemoveWorkflowAssociationSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="ConvertWebPartFormat">
               <wsdl:input message="tns:ConvertWebPartFormatSoapIn" />
               <wsdl:output message="tns:ConvertWebPartFormatSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetAssemblyMetaData">
               <wsdl:input message="tns:GetAssemblyMetaDataSoapIn" />
               <wsdl:output message="tns:GetAssemblyMetaDataSoapOut" />
            </wsdl:operation>
            <wsdl:operation name="GetBindingResourceData">
```

```
            <wsdl:input message="tns:GetBindingResourceDataSoapIn" />
            <wsdl:output message="tns:GetBindingResourceDataSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="ExecuteProxyUpdates">
            <wsdl:input message="tns:ExecuteProxyUpdatesSoapIn" />
            <wsdl:output message="tns:ExecuteProxyUpdatesSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="AddWebPartToZone">
            <wsdl:input message="tns:AddWebPartToZoneSoapIn" />
            <wsdl:output message="tns:AddWebPartToZoneSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="AddWebPart">
            <wsdl:input message="tns:AddWebPartSoapIn" />
            <wsdl:output message="tns:AddWebPartSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPart">
            <wsdl:input message="tns:GetWebPartSoapIn" />
            <wsdl:output message="tns:GetWebPartSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPart2">
            <wsdl:input message="tns:GetWebPart2SoapIn" />
            <wsdl:output message="tns:GetWebPart2SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetCustomControlList">
            <wsdl:input message="tns:GetCustomControlListSoapIn" />
            <wsdl:output message="tns:GetCustomControlListSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetSafeAssemblyInfo">
            <wsdl:input message="tns:GetSafeAssemblyInfoSoapIn" />
            <wsdl:output message="tns:GetSafeAssemblyInfoSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPartProperties">
            <wsdl:input message="tns:GetWebPartPropertiesSoapIn" />
            <wsdl:output message="tns:GetWebPartPropertiesSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPartProperties2">
            <wsdl:input message="tns:GetWebPartProperties2SoapIn" />
            <wsdl:output message="tns:GetWebPartProperties2SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart">
            <wsdl:input message="tns:SaveWebPartSoapIn" />
            <wsdl:output message="tns:SaveWebPartSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart2">
            <wsdl:input message="tns:SaveWebPart2SoapIn" />
            <wsdl:output message="tns:SaveWebPart2SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="DeleteWebPart">
            <wsdl:input message="tns:DeleteWebPartSoapIn" />
            <wsdl:output message="tns:DeleteWebPartSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPartPageConnectionInfo">
            <wsdl:input message="tns:GetWebPartPageConnectionInfoSoapIn" />
            <wsdl:output message="tns:GetWebPartPageConnectionInfoSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetWebPartCrossPageCompatibility">
            <wsdl:input message="tns:GetWebPartCrossPageCompatibilitySoapIn" />
            <wsdl:output message="tns:GetWebPartCrossPageCompatibilitySoapOut" />
        </wsdl:operation>
```

```
    </wsdl:portType>
    <wsdl:binding name="WebPartPagesWebServiceSoap" type="tns:WebPartPagesWebServiceSoap">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsdl:operation name="GetWebPartPageDocument">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPageDocument"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartPage">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPage" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="RenderWebPartForEdit">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/RenderWebPartForEdit"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetXmlDataFromDataSource">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetXmlDataFromDataSource"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetFormCapabilityFromDataSourceControl">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetFormCapabilityFromDataSourceContr
ol" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetDataFromDataSourceControl">
```

```
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetDataFromDataSourceControl"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetExpandedListViewXml">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetExpandedListViewXml"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="FetchLegalWorkflowActions">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/FetchLegalWorkflowActions"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="ValidateWorkflowMarkupAndCreateSupportObjects">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ValidateWorkflowMarkupAndCreateSuppo
rtObjects" style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="AssociateWorkflowMarkup">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/AssociateWorkflowMarkup"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="RemoveWorkflowAssociation">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/RemoveWorkflowAssociation"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
```

```
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="ConvertWebPartFormat">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ConvertWebPartFormat"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetAssemblyMetaData">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetAssemblyMetaData"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetBindingResourceData">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetBindingResourceData"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="ExecuteProxyUpdates">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ExecuteProxyUpdates"
style="document" />
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="AddWebPartToZone">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/AddWebPartToZone" style="document"
/>
            <wsdl:input>
               <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
```

```
      <wsdl:operation name="AddWebPart">
         <soap:operation soapAction="http://microsoft.com/sharepoint/webpartpages/AddWebPart"
style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetWebPart">
         <soap:operation soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPart"
style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetWebPart2">
         <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPart2" style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetCustomControlList">
         <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetCustomControlList"
style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetSafeAssemblyInfo">
         <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetSafeAssemblyInfo"
style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetWebPartProperties">
         <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties"
style="document" />
         <wsdl:input>
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output>
```

```
                    <soap:body use="literal" />
                </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartProperties2">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties2"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/SaveWebPart" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart2">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/SaveWebPart2" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="DeleteWebPart">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/DeleteWebPart" style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartPageConnectionInfo">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPageConnectionInfo"
style="document" />
            <wsdl:input>
                <soap:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartCrossPageCompatibility">
            <soap:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartCrossPageCompatibility"
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="WebPartPagesWebServiceSoap12" type="tns:WebPartPagesWebServiceSoap">
            <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
            <wsdl:operation name="GetWebPartPageDocument">
               <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPageDocument"
style="document" />
               <wsdl:input>
                  <soap12:body use="literal" />
               </wsdl:input>
               <wsdl:output>
                  <soap12:body use="literal" />
               </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="GetWebPartPage">
               <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPage" style="document" />
               <wsdl:input>
                  <soap12:body use="literal" />
               </wsdl:input>
               <wsdl:output>
                  <soap12:body use="literal" />
               </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="RenderWebPartForEdit">
               <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/RenderWebPartForEdit"
style="document" />
               <wsdl:input>
                  <soap12:body use="literal" />
               </wsdl:input>
               <wsdl:output>
                  <soap12:body use="literal" />
               </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="GetXmlDataFromDataSource">
               <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetXmlDataFromDataSource"
style="document" />
               <wsdl:input>
                  <soap12:body use="literal" />
               </wsdl:input>
               <wsdl:output>
                  <soap12:body use="literal" />
               </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="GetFormCapabilityFromDataSourceControl">
               <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetFormCapabilityFromDataSourceContr
ol" style="document" />
               <wsdl:input>
                  <soap12:body use="literal" />
               </wsdl:input>
               <wsdl:output>
```

```
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetDataFromDataSourceControl">
           <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetDataFromDataSourceControl"
style="document" />
           <wsdl:input>
              <soap12:body use="literal" />
           </wsdl:input>
           <wsdl:output>
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetExpandedListViewXml">
           <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetExpandedListViewXml"
style="document" />
           <wsdl:input>
              <soap12:body use="literal" />
           </wsdl:input>
           <wsdl:output>
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="FetchLegalWorkflowActions">
           <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/FetchLegalWorkflowActions"
style="document" />
           <wsdl:input>
              <soap12:body use="literal" />
           </wsdl:input>
           <wsdl:output>
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ValidateWorkflowMarkupAndCreateSupportObjects">
           <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ValidateWorkflowMarkupAndCreateSuppo
rtObjects" style="document" />
           <wsdl:input>
              <soap12:body use="literal" />
           </wsdl:input>
           <wsdl:output>
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="AssociateWorkflowMarkup">
           <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/AssociateWorkflowMarkup"
style="document" />
           <wsdl:input>
              <soap12:body use="literal" />
           </wsdl:input>
           <wsdl:output>
              <soap12:body use="literal" />
           </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="RemoveWorkflowAssociation">
```

```
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/RemoveWorkflowAssociation"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ConvertWebPartFormat">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ConvertWebPartFormat"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetAssemblyMetaData">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetAssemblyMetaData"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetBindingResourceData">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetBindingResourceData"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ExecuteProxyUpdates">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/ExecuteProxyUpdates"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="AddWebPartToZone">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/AddWebPartToZone" style="document"
/>
            <wsdl:input>
                <soap12:body use="literal" />
```

```
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="AddWebPart">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/AddWebPart" style="document" />
            <wsdl:input>
               <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetWebPart">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPart" style="document" />
            <wsdl:input>
               <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetWebPart2">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPart2" style="document" />
            <wsdl:input>
               <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetCustomControlList">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetCustomControlList"
style="document" />
            <wsdl:input>
               <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetSafeAssemblyInfo">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetSafeAssemblyInfo"
style="document" />
            <wsdl:input>
               <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
               <soap12:body use="literal" />
            </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetWebPartProperties">
```

```
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartProperties2">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartProperties2"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/SaveWebPart" style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="SaveWebPart2">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/SaveWebPart2" style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="DeleteWebPart">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/DeleteWebPart" style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetWebPartPageConnectionInfo">
            <soap12:operation
soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartPageConnectionInfo"
style="document" />
            <wsdl:input>
                <soap12:body use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal" />
```

```
              </wsdl:output>
         </wsdl:operation>
         <wsdl:operation name="GetWebPartCrossPageCompatibility">
            <soap12:operation
 soapAction="http://microsoft.com/sharepoint/webpartpages/GetWebPartCrossPageCompatibility"
 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 2010 suites

- Microsoft® Office 2013 Preview

- Microsoft® SharePoint® Designer 2010

- Microsoft® SharePoint® Designer 2013 Preview

- Windows® SharePoint® Services 2.0

- Windows® SharePoint® Services 3.0

- Microsoft® SharePoint® Foundation 2010

- Microsoft® SharePoint® Foundation 2013 Preview

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

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

<1> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface specified is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface specified is of the type described in [MSDN-WCCSWP].

<2> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface is of the type described in [MSDN-WCCSWP].

<3> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface is of the type described in [MSDN-WCCSWP].

<4> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface is of the type described in [MSDN-WCCSWP].

<5> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface is of the type described in [MSDN-WCCSWP].

<6> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the **IsASP** property is set to TRUE, the Web Part connection interface is of the type described in [MSDN-WPCO]. If the **IsASP** property is set to FALSE, the Web Part connection interface is of the type described in [MSDN-WCCSWP].

<7> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, each Web Part connection transformation implements its own format for the **XML.ConnDesign.Compatibility.sWebPart.sg.tParts.tWebPart.tg.mi.xInfo** element. For example, the Microsoft.SharePoint.WebPartPages.Communication.RowFilterTransform uses a format of a single string matching ([^=]+)=([^=]+) where the first matching group (before the equals sign) is the name of the field to filter on, and the second matching group (after the equals sign) is the value to filter on. In contrast, the Microsoft.SharePoint.WebPartPages.Communication.RowParametersInTransform also uses a single string, but this matches (\(([^=\(\)])+=([^=\(\)])+\))+, where there are repeated groups separated by parenthesis that are formatted such that, within each of those repeated groups, each part that is before an equals sign is a name of a parameter and each part after an equals sign is the value of that parameter.

<8> Section 2.2.3.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this is a fully-qualified type name deriving from **System.Web.UI.WebControls.WebParts.WebPartTransformer**. For example: System.Web.UI.WebControls.WebParts.RowToParametersTransformer, System.Web, Version=2.0.0.0, Culture=neutral, PublicKeyToken= b03f5f7f11d50a3a.

<9> Section 2.2.4.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the additional attributes are namespace declarations.

<10> Section 2.2.4.3: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this is an assembly-qualified .NET type name, as described in [MSDN-WPCDF].

<11> Section 2.2.4.3: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this is the path to a user control, as described in [MSDN-WPCDF].

<12> Section 2.2.4.3: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this is either an assembly-qualified .NET type name or one of the attribute values described in [MSDN-WPCDF].

<13> Section 2.2.4.4: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the implementation is specific to each Web Part connection interface type. For example, the Micrososft.SharePoint.WebPartPages.Communication.IListProvider uses XML of the form:

```
<s:element name="ListProviderInitEventArgs">
    <s:complexType>
       <s:sequence>
          <s:element name="Field" minOccurs="0" maxOccurs="unbounded">
             <s:complexType>
                <s:sequence />
                 <s:attribute name="FieldName" type="s:string" />
                 <s:attribute name="FieldDisplayName" type="s:string" />
             </s:complexType>
          </s:element>
       </s:sequence>
    </s:complexType>
</s:element>
```

Wherease the Micrososft.SharePoint.WebPartPages.Communication.IParametersInProvider interface uses XML of the form:

```
<s:element name="ParametersInConsumerInitEventArgs">
   <s:complexType>
      <s:sequence>
         <s:element name="Parameter" minOccurs="0" maxOccurs="unbounded">
            <s:complexType>
               <s:sequence />
               <s:attribute name="ParameterName" type="s:string" />
               <s:attribute name="ParameterDisplayName" type="s:string" />
               <s:attribute name="Description" type="s:string" />
               <s:attribute name="Required" type="s:boolean" />

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

Each implementation of different Web Part connection interfaces has its own schema for the contents of **InitEventArgsType.***.

<14> Section 2.2.5.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if the storage value is set to "None", this is interpreted as "Personal" when reading, updating, and deleting Web Parts, and causes an error when adding Web Parts.

<15> Section 3.1.4.2: Windows SharePoint Services 2.0 does not support this operation.

<16> Section 3.1.4.2.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the DWP format and the format described in [MSDN-WPCDF] are supported.

<17> Section 3.1.4.2.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the value of **AddWebPartToZone.zoneId** corresponds to the **ID** property of a **WebPartZone** control in the page.

<18> Section 3.1.4.2.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, Web Parts that have no valid zone are kept in the database, but are not rendered on the page. They can later be moved to a valid Web Part zone on the page once one is created.

<19> Section 3.1.4.2.2.1: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 assign a new Web Part zone index to Web Parts when a user first browses to a page that includes a Web Part with invalid or duplicate indices.

<20> Section 3.1.4.3: Windows SharePoint Services 2.0 does not support this operation.

<21> Section 3.1.4.3: The syntax of the configuration file used in the implementation of this operation in Windows SharePoint Services 3.0 and SharePoint Foundation 2010 is described in [MSDN-WCSO] as follows:

```
<s:schema elementFormDefault="qualified">
   <s:element name="WorkflowConfig">
      <s:complexType>
         <s:sequence>
            <s:element name="Template" minOccurs="0" maxOccurs="1">
               <s:complexType>
```

```
                     <s:attribute name="AllowStartOnMajorCheckin" type="s:boolean"
use="optional" />
                     <s:attribute name="AssociationURL" type="s:string" use="optional" />
                     <s:attribute name="BaseID" type="s:string" use="optional"/>
                     <s:attribute name="Category" type="s:string" use="optional" />
                     <s:attribute name="Description" type="s:string" use="optional" />
                     <s:attribute name="DocLibID" type="s:string" use="optional"/>
                     <s:attribute name="DocLibURL" type="s:string" use="optional" />
                     <s:attribute name="PreviewHref" type="s:string" use="optional" />
                     <s:attribute name="PreviewRequiresGeneration" type="s:string"
default="false" use="optional" />
                     <s:attribute name="PreviewVersion" type="s:string" use="optional" />
                     <s:attribute name="RequireManagePermission" type="s:boolean" use="optional"
/>
                     <s:attribute name="RestrictLanguage" type="s:integer" use="optional" />
                     <s:attribute name="Ruleset" type="s:string" use="optional" />
                     <s:attribute name="RulesVersion" type="s:string" use="optional"/>
                     <s:attribute name="RulesHref" type="s:string" use="optional" />
                     <s:attribute name="ShowPreview" type="s:boolean" use="optional" />
                     <s:attribute name="StartManually" type="s:boolean" use="optional" />
                     <s:attribute name="StartOnChange" type="s:boolean" use="optional" />
                     <s:attribute name="StartOnCreate" type="s:boolean" use="optional" />
                     <s:attribute name="TaskContentType" type="s:string" use="optional" />
                     <s:attribute name="Type">
                        <s:simpleType>
                           <s:restriction base="s:string">
                              <s:enumeration value="RulesOnly" />
                           </s:restriction>
                        </s:simpleType>
                     </s:attribute>
                     <s:attribute name="Visibility" use="optional">
                        <s:simpleType>
                           <s:restriction base="s:string">
                              <s:enumeration value="RootPublic" />
                              <s:enumeration value="Public" />
                              <s:enumeration value="DraftRootPublic" />
                              <s:enumeration value="DraftPublic" />
                           </s:restriction>
                        </s:simpleType>
                     </s:attribute>
                     <s:attribute name="XomlVersion" type="s:string" use="optional"/>
                     <s:attribute name="XomlHref" type="s:string" use="optional"/>
                  </s:complexType>
               </s:element>
               <s:element name="Association" minOccurs="0" maxOccurs="1">
                  <s:complexType>
                     <s:sequence>
                     </s:sequence>
                     <s:attribute name="ListID" type="s:string" use="optional" />
                     <s:attribute name="TaskListID" type="s:string" use="optional" />
                     <s:attribute name="HistoryListID" type="s:string" use="optional" />
                     <s:attribute name="RequireManagePermission" type="s:boolean" use="optional"
/>
                     <s:attribute name="StartManually" type="s:boolean" use="optional" />
                     <s:attribute name="StartOnCreate" type="s:boolean" use="optional" />
                     <s:attribute name="StartOnChange" type="s:boolean" use="optional" />
                  </s:complexType>
               </s:element>
               <s:element name="ContentTypes" minOccurs="0" maxOccurs="1">
```

198 / 224

```
                    <s:complexType>
                       <s:sequence>
                          <s:element name="ContentType" minOccurs="0" maxOccurs="unbounded">
                             <s:complexType>
                                <s:sequence>
                                   <s:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                                </s:sequence>
                                <s:attribute name="ContentTypeID" type="s:string" use="required"
/>
                                <s:attribute name="RequiresGeneration" type="s:boolean"
default="false" use="optional"/>
                                <s:attribute name="Form" type="s:string" />
                                <s:attribute name="WorkflowForm" type="s:string" />
                                <s:anyAttribute namespace="##any" processContents="lax"/>
                             </s:complexType>
                          </s:element>
                       </s:sequence>
                    </s:complexType>
                 </s:element>
                 <s:element name="Initiation" minOccurs="0" maxOccurs="1">
                    <s:complexType>
                       <s:sequence>
                          <s:element name="Fields" minOccurs="0" maxOccurs="1">
                             <s:complexType>
                                <s:sequence>
                                   <s:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                                </s:sequence>
                             </s:complexType>
                          </s:element>
                          <s:element name="Parameters" minOccurs="0" maxOccurs="1">
                             <s:complexType>
                                <s:sequence>
                                   <s:element name="Parameter" minOccurs="0"
maxOccurs="unbounded">
                                      <s:complexType>
                                         <s:attribute name="Type" type="s:string" use="required"
/>
                                         <s:attribute name="Name" type="s:string" use="required"
/>
                                      </s:complexType>
                                   </s:element>
                                </s:sequence>
                             </s:complexType>
                          </s:element>
                          <s:element name="DefaultData" minOccurs="0" maxOccurs="unbounded">
                             <s:complexType>
                                <s:sequence>
                                   <s:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                                </s:sequence>
                             </s:complexType>
                          </s:element>
                       </s:sequence>
                       <s:attribute name="URL" type="s:string" use="optional" />
                    </s:complexType>
                 </s:element>
                 <s:element name="Extended" minOccurs="0" maxOccurs="1">
                    <s:complexType>
```

```
                                 <s:sequence>
                                    <s:element name="Fields" minOccurs="0" maxOccurs="1">
                                       <s:complexType>
                                          <s:sequence>
                                             <s:element name="Field" type="core:AddOrUpdateFieldDefinition"
 minOccurs="0" maxOccurs="unbounded"/>
                                          </s:sequence>
                                       </s:complexType>
                                    </s:element>
                                 </s:sequence>
                              </s:complexType>
                           </s:element>
                           <s:element name="MetaData" minOccurs="0" maxOccurs="1">
                              <s:complexType>
                                 <s:all>
                                    <s:element name="TaskForms" minOccurs="0" maxOccurs="1">
                                       <s:complexType>
                                          <s:sequence>
                                             <s:element name="TaskForm" minOccurs="0" maxOccurs="unbounded">
                                                <s:complexType>
                                                   <s:sequence>
                                                   </s:sequence>
                                                   <s:attribute name="TaskType" type="s:integer" />
                                                   <s:attribute name="FormUrn" type="s:string" />
                                                </s:complexType>
                                             </s:element>
                                          </s:sequence>
                                       </s:complexType>
                                    </s:element>
                                    <s:element name="Modifications" minOccurs="0" maxOccurs="1">
                                       <s:complexType>
                                          <s:sequence>
                                             <s:element name="Modification" minOccurs="1"
 maxOccurs="unbounded">
                                                <s:complexType>
                                                   <s:sequence>
                                                   </s:sequence>
                                                   <s:attribute name="TypeId" type="s:string"
 use="required"/>
                                                   <s:attribute name="NameFormat" type="s:string"
 use="required"/>
                                                   <s:attribute name="FormURN" type="s:string"
 use="required"/>
                                                </s:complexType>
                                             </s:element>
                                          </s:sequence>
                                          <s:attribute name="ModificationUrl" type="s:string"
 use="required"/>
                                       </s:complexType>
                                    </s:element>
                                    <s:element name="Instantiation_FormURI" type="s:string" minOccurs="0"
 maxOccurs="1">
                                       <s:complexType mixed="true">
                                          <s:sequence />
                                          <s:attribute name="RequiresGeneration" type="s:boolean"
 default="false" use="optional" />
                                          <s:attribute name="Version" default="v1.0" use="optional">
                                             <s:restriction base="s:string">
                                                <s:pattern value="[v|V]\d+\.\d+" />
                                             </s:restriction>
```

```
                            </s:attribute>

                        </s:complexType>
                    </s:element>
                    <s:element name="Association_FormURN" type="s:string" minOccurs="0"
   maxOccurs="1">
                        <s:complexType mixed="true">
                            <s:sequence />
                            <s:attribute name="RequiresGeneration" type="s:boolean"
   default="false" use="optional" />
                            <s:attribute name="Version" default="v1.0" use="optional">
                                <s:restriction base="s:string">
                                    <s:pattern value="[v|V]\d+\.\d+" />
                                </s:restriction>
                            </s:attribute>
                        </s:complexType>
                    </s:element>
                    <s:element name="SigClientSettings" type="s:string" minOccurs="0"
   maxOccurs="1"/>
                    <s:element name="HasStatusColumn" type="s:boolean" minOccurs="0"
   maxOccurs="1"/>
                    <s:element name="HasTaskList" type="s:boolean" minOccurs="0"
   maxOccurs="1"/>
                    <s:element name="HasHistoryList" type="s:boolean" minOccurs="0"
   maxOccurs="1"/>
                    <s:element name="ExtendedStatusColumnValues" minOccurs="0"
   maxOccurs="1">
                        <s:complexType>
                            <s:sequence>
                                <s:element name="StatusColumnValue" type="s:string"
   minOccurs="0" maxOccurs="unbounded" />
                            </s:sequence>
                        </s:complexType>
                    </s:element>
                </s:all>
            </s:complexType>
        </s:element>
        <s:element name="AssociateOnActivation" type="s:boolean" minOccurs="0"
   maxOccurs="1"/>
      </s:sequence>
      <s:attribute name="Version" type="s:string" />
    </s:complexType>
  </s:element>
</s:schema>
```

WorkflowConfig: Root element of the workflow configuration file.

WorkflowConfig.Template: Specifies information about the location of the workflow markup and invariant information about the workflow.

WorkflowConfig.Template.DocLibID: The GUID of the document library that contains the workflow markup.

WorkflowConfig.Template.DocLibURL: The URL of the document library that contains the workflow markup. If **docLibID** is specified, then this value is ignored.

WorkflowConfig.Template.BaseID: The GUID that uniquely identifies the workflow. This value is invariant across updates to the workflow.

WorkflowConfig.Template.XomlHref: The site-relative URL of the workflow **Extensible Application Markup Language (XAML)** file.

WorkflowConfig.Template.XomlVersion: The version number of the workflow XAML file at the location specified in **XomlHref**.

WorkflowConfig.Template.RulesHref: The site-relative URL of the workflow rules file.

WorkflowConfig.Template.RulesVersion: The version number of the workflow rules file specified by **RulesHref**.

WorkflowConfig.Template.Category: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.Description: A textual description available for the server to show in UI elements that gives the user information about what the workflow is expected to do. This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Template.ShowPreview: If set to TRUE, the server will enable, if possible, the current state of the workflow in a diagram when the user accesses the current state of the workflow in the UI. This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Template.PreviewHref: If this value begins with a slash (/), it refers to a site-collection-relative URL. If this value does not begin with slash, it refers to a site-relative URL. If set and **ShowPreview** is set to TRUE, the server will, if possible, render the diagram located at **PreviewHref** to show the current state of the workflow when the user accesses the current state of the workflow through the UI. This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Template.PreviewVersion: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.PreviewRequiresGeneration: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.Type: If this is set to "RulesOnly", this workflow does not use Extensible Application Markup Language (XAML) to define the execution of the workflow. It uses the workflow rules file specified by **RulesHref**. If this is set to "RulesOnly", **Ruleset** is set. This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Template.Ruleset: If set, this specifies which rule in the workflow rules file to evaluate when the workflow is executed. If it is set, **Type** is set to "RulesOnly". This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Template.Visibility: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.AssociationURL: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.AllowStartOnMajorCheckin: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.StartManually: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.StartOnChange: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.StartOnCreate: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.RequireManagePermission: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.RestrictLanguage: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Template.TaskContentType: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Association: Information about how the client authoring the workflow can associate the workflow with objects on the server.

WorkflowConfig.Association.ListID: The GUID of the list (1) with which the workflow is associated.

WorkflowConfig.Association.TaskListID: The GUID of a list (1) with different **content types** applied to it. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if this attribute is empty or null, the server selects an existing task list (1) or creates a new task list (1) with a default name. If the server either reuses an existing task list (1) or creates a new task list (1), all future references to **WorkflowConfig.Association.TaskListID** use the value of the GUID identifier of the list (1). See also **HasTaskList.**

WorkflowConfig.Association.StartManually: If set to TRUE, indicates that the server makes the workflow available for starting manually through UI elements. If set to FALSE, indicates that the server does not make the workflow available for starting manually through UI elements.

WorkflowConfig.Association.StartOnChange: If set to TRUE, indicates that the server starts the workflow when a list item or **document** in the **WorkflowConfig.Association.ListID** list (1) or document library is modified. If set to FALSE, indicates that the server does not start the workflow when a list item or document in the **WorkflowConfig.Association.ListID** list (1) or document library is modified.

WorkflowConfig.Association.StartOnCreate: If set to TRUE, indicates that the server starts the workflow when a list item or document in the **WorkflowConfig.Association.ListID** list (1) or document library is created. If set to FALSE, indicates that the server does not start the workflow when a list item or document in the **WorkflowConfig.Association.ListID** list (1) or document library is created.

WorkflowConfig.Association.HistoryListID: The GUID of a list (1) in which to place list items that record information about the workflow. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if this attribute is empty or null, the server selects an existing list (1) or creates a new list (1) with a default name. If the server either reuses an existing list (1) or creates a new list (1), all future references to **WorkflowConfig.Association.HistoryListID** use the GUID for the list (1). See also **HasHistoryList**.

WorkflowConfig.Association.RequireManagePermission: If set to TRUE, the server requires additional permissions for a user to start the workflow through UI elements. If set to FALSE, the server does not require additional permissions for a user to start the workflow through UI elements. This attribute is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.ContentTypes: This element is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.ContentTypes.ContentType: This element is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.ContentTypes.ContentType.ContentTypeID: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.ContentTypes.ContentType.Form This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.ContentTypes.ContentType.WorkflowForm This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.Initiation: Specifies information for processing a form used to start a workflow.

WorkflowConfig.Initiation.URL: Specifies the site-relative URL of the initiation form.

WorkflowConfig.Initiation.Parameters: Specifies the set of parameters needed to successfully start the workflow.

WorkflowConfig.Initiation.Parameters.Parameter: Specifies a single parameter needed for starting the workflow.

WorkflowConfig.Initiation.Parameters.Parameter.Name: The name of the parameter.

WorkflowConfig.Initiation.Parameters.Parameter.Type: The .NET type of the parameter, for example: "System.String".

WorkflowConfig.Initiation.Fields: Specifies the set of fields that appear in the initiation form.

WorkflowConfig.Initiation.Fields.Field: The standard **Field** elements of the **AddOrUpdateFieldDefinition** type, as described in [MS-LISTSWS], Section 2.2.4.1.

In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, there are only two pieces of information used by the server: the **Name** attribute of the **Field** element and the value of the **Default** child element.

When constructing the name-value pairs to pass to the workflow at initiation, the server builds a set of name-value pairs from the default values of each **Field** element, merges this with name-value pairs from the workflow initiation form, and then converts each value to the .NET type, as specified by a **WorkflowConfig.Initiation.Parameters.Parameter** element in which the name of the **Parameter** element exactly matches the name of the **Field** element.

If the **DefaultData** element is present, the server uses the contents of that element, instead of following the algorithm outlined previously.

WorkflowConfig.Initiaiton.DefaultData: If set, the server uses the contents of this element for the data that is described in the **WorkflowConfig.Initaition.Fields.Field** element. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Extended.Fields: The contents of the **Fields** element are standard **Field** elements of the **AddOrUpdateFieldDefinition** type, as described in [MS-LISTSWS], Section 2.2.4.1. When a workflow association is created for a list (1) or document library, the server adds each of these **Field** elements to that list (1) or document library. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData: The child elements of this element specify various additional properties of a workflow. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.TaskForms: Each **TaskForm** child element describes one form used to modify a workflow task and how an integer refers to the form. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.TaskForms.TaskForm: Describes how an integer used in a workflow can refer to a task form that has a URN. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.TaskForms.TaskForm.TaskType: Specifies the integer used to refer to a task form. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.TaskForms.TaskForm.FormUrn: Specifies the URN of the form to modify the task. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications: Each child **Modification** element specifies a workflow modification. A workflow modification is a form in the UI that enables the user to pass data directly to the workflow. The internal workflow logic specifically activates each modification and could deactivate the modification during the lifetime of the workflow. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications.Modification: Specifies one workflow modification. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications.Modification.TypeId: Specifies the identifier that the workflow uses to activate and deactivate the workflow modification. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications.Modification.NameFormat: Specifies the text displayed to users to begin interaction with the workflow modification. If the client places the token {0} in the string, the internal workflow logic can replace it with implementation-specific text. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications.Modification.FormURN: Specifies the URN of the form to display to the user when the user is interacting with the workflow modification. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Modifications.ModificationUrl: Specifies the URL of a form that loads the workflow modification by the URN of the workflow modification. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Instantiation_FormURI: Specifies the URN of the form to display to the user to start the workflow. The server can use this value in conjunction with **Initiation.URL**. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Instantiation_FormURI.RequiresGeneration: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.MetaData.Instantiation_FormURI.Version: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.MetaData.Association_FormURN: This element is ignored for the **AssociateWorkflowMarkup** request. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.Association_FormURN.RequiresGeneration: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.MetaData.Association_FormURN.Version: This attribute is ignored for the **AssociateWorkflowMarkup** request.

WorkflowConfig.MetaData.SigClientSettings: Specifies additional properties that be returned to the client in subsequent calls to the **WorkflowTemplates.WorkflowTemplate.Metadata.SigClientSettings** element**.** This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.HasStatusColumn: If set to TRUE, the server provide a field of data on the list item or document on which the workflow is running that indicates the current status of the workflow. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.ExtendedStatusColumnValues: Each child **StatusColumnValue** element specifies a value that indicates a possible current status of the workflow. These are in addition to any values that the server implementation provides. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.ExtendedStatusColumnValues.StatusColumnValue: Specifies one value for which the internal workflow logic can choose to set the indicator for the list item or document on which the workflow is running to create user-accessible information about the current status of the workflow. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.HasTaskList: If set to TRUE, this workflow has a list (1) to which to add list items that track work that is assigned to users of the system. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.HasHistoryList: If set to TRUE, this workflow has a list (1) to which to add list items that describe the execution of the workflow. The server enable the workflow to add items to this list (1), and the server can also add items to this list (1). This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.MetaData.AssociateOnActivation: If set to TRUE, this workflow is available to be activated when a **feature** containing the workflow is activated. This element is ignored in Windows SharePoint Services 3.0.

WorkflowConfig.Version: Specifies which version of the server was in use when the Extensible Application Markup Language (XAML) was last modified. This attribute is ignored in Windows SharePoint Services 3.0.

<22> Section 3.1.4.4: Windows SharePoint Services 2.0 does not support this operation.

<23> Section 3.1.4.4: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, when **formatConversionOption** is set to "ConvertToWebPartDesignerPersistenceFormat", the input format expected is the DWP format defined in **wpv2:WebPart** (section 2.2.4.2) or the .webpart format defined in webParts (section 2.2.3.1), and the output format is ASP.NET page markup. When **formatConversionOption** is set to "ConvertToWebPartExportFormat", the input format expected is ASP.NET page markup, and the output format is the DWP format specified in **wpv2:WebPart** (section 2.2.4.2) or the .webpart format defined in webParts (section 2.2.3.1), depending on the Web Part.

<24> Section 3.1.4.6: Windows SharePoint Services 2.0 does not support this operation.

<25> Section 3.1.4.6.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the controlType.TagName is a prefixed ASP.NET tag name, such as example:value, where example corresponds to a register directive with a matching tagPrefix, and value corresponds to a type located in a way findable with register directives. See [MSDN-Register] for more details on register directives.

<26> Section 3.1.4.6.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **UpdateResponse.Update.Method.N** attribute is the method name of any public method available on the ASP.NET Control Designer (System.Web.UI.Design.ControlDesigner) for the control that is referenced by the **UpdateID** attribute on the same **UpdateResponse.Update.Method** element.

<27> Section 3.1.4.7: Windows SharePoint Services 2.0 does not support this operation.

<28> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **ItemIdType** is either System.Int32 or Microsoft.SharePoint.Workflow.SPItemKey.

<29> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **FunctionName** attribute specifies the name of a static Boolean function of the class specified by the **ClassName** attribute.

<30> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<31> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **ClassName** attribute contains a fully-qualified namespace and class name.

<32> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **ItemIdType** is either System.Int32 or Microsoft.SharePoint.Workflow.SPItemKey.

<33> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<34> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **ClassName** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with Windows SharePoint Services 3.0 and SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<35> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, this conforms to the schema, as follows:

```
<s:schema targetNamespace="clr-namespace:Microsoft.SharePoint
.WorkflowActions;Assembly=Microsoft.SharePoint.WorkflowActions, Version=14.0.0.0,
Culture=neutral, PublicKeyToken=null">
   <s:element name="WorkflowDataFields" minOccurs="1" maxOccurs="1">
      <s:complexType>
         <s:sequence>
            <s:element name="WorkflowDataField" minOccurs="0" maxOccurs="unbounded">
               <s:complexType>
                  <s:sequence />
                  <s:attribute name="Name" type="s:string" use="required" />
                  <s:attribute name="Type" type="s:string" use="required" />
               </s:complexType>
            </s:element>
         </s:sequence>
      </s:complexType>
   </s:element>
</s:schema>
```

WorkflowDataFields: Specifies a list of local variables to add to the workflow if the ancestor action (2) is added to the internal workflow logic.

WorkflowDataFields.WorkflowDataField: Specifies one local variable to add to the workflow if the ancestor action (2) is added to the internal workflow logic.

WorkflowDataFields.WorkflowDataField.Name: Specifies the name of the local variable to add to the workflow.

WorkflowDataFields.WorkflowDataField.Type: Specifies the class type of the local variable to add to the workflow.

<36> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, this content is equivalent to a portion of Extensible Application Markup Language (XAML).

<37> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, this content is equivalent to a portion of a workflow rules file.

<38> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, this action (2) is **Microsoft.SharePoint.WorkflowActions.SPUserCodeWorkflowActivity**.

<39> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **FunctionName** attribute specifies the name of a static function of the class specified by the **ClassName** attribute.

<40> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, this action (2) is **Microsoft.SharePoint.WorkflowActions.SPUserCodeWorkflowActivity**.

<41> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<42> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<43> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<44> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<45> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **ClassName** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<46> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<47> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<48> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<49> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<50> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **Type** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<51> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<52> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<53> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<54> Section 3.1.4.7.2.2: This attribute is not returned in Windows SharePoint Services 3.0.

<55> Section 3.1.4.7.2.2: This attribute is not returned in Windows SharePoint Services 3.0.

<56> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<57> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **From** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<58> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<59> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **To** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<60> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<61> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<62> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **Assembly** attribute contains the fully-qualified name of the assembly, in .NET assembly reference syntax.

<63> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<64> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **ClassName** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<65> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **SourceType** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<66> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **DestinationType** attribute contains a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<67> Section 3.1.4.7.2.2: In SharePoint Foundation 2010, the **string** value to replace the placeholder with is a fully-qualified namespace and class name. For information about creating classes that are compatible with SharePoint Foundation 2010, see [MSDN-BCAUSPD].

<68> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, **TypeFilter** is a semicolon-delimited list of namespace-qualified .NET type names. For example, the operator "contains the substring" is appropriate only if the left-hand operand is a string, so the **TypeFilter** attribute for that option would be set to "System.String", and thus the "contains the substring" operator would only appear in the list of available options for a **FieldBind** if the value of the parameter specified in the **Field** attribute of that **FieldBind** was of type "System.String".

<69> Section 3.1.4.7.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Type** attribute contains a fully-qualified .NET namespace, type, and assembly name string, for example, "System.String, mscorlib".

<70> Section 3.1.4.7.2.2: This attribute is not returned in Windows SharePoint Services 3.0.

<71> Section 3.1.4.7.2.2: This attribute is not returned in Windows SharePoint Services 3.0.

<72> Section 3.1.4.7.2.2: This attribute is not returned in Windows SharePoint Services 3.0.

<73> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<74> Section 3.1.4.7.2.2: This element is not returned in Windows SharePoint Services 3.0.

<75> Section 3.1.4.8: Windows SharePoint Services 2.0 does not support this operation.

<76> Section 3.1.4.8.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010. **assemblyName** is the fully-qualified name of the assembly, in .NET assembly reference syntax.

<77> Section 3.1.4.8.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, **baseTypes** is a list of .NET assembly qualified class names.

<78> Section 3.1.4.8.2.2: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 return an XML-encoded string that contains the XML of the following schema. Each type that is referred to or specified in this XML has a non-negative integer assigned to it.

```
<s:schema elementFormDefault="qualified">
  <s:complexType name="AType">
    <s:sequence>
      <s:element name="G" type="GType" minOccurs="0" maxOccurs="unbounded" />
    </s:sequence>
    <s:attribute name="Y" type="s:string" use="required"/>
  </s:complexType>

  <s:complexType name="EType">
    <s:sequence>
      <s:element name="Value" minOccurs="0" maxOccurs="unbounded">
        <s:complexType>
          <s:simpleContent>
            <s:extension base="s:string">
              <s:attribute name="N" type="s:string" />
            </s:extension>
          </s:simpleContent>
        </s:complexType>
      </s:element>
    </s:sequence>
    <s:attribute name="ID" type="s:nonNegativeInteger" use="required"/>
    <s:attribute name="Y" type="s:nonNegativeInteger" use="optional"/>
  </s:complexType>

  <s:complexType name="GType">
    <s:sequence />
    <s:attribute name="Y" type="s:string" use="required"/>
    <s:attribute name="ID" type="s:nonNegativeInteger" use="optional"/>
    <s:attribute name="E" type="s:string" use="optional"/>
    <s:attribute name="Serial" type="s:string"  use="optional"/>
    <s:attribute name="T" type="s:string"  use="optional"/>
    <s:attribute name="Bound" type="s:string" use="optional"/>
    <s:attribute name="AssemblyQualifiedName" type="s:string" use="optional" />
  </s:complexType>

  <s:complexType name="PType">
    <s:sequence>
      <s:choice maxOccurs="unbounded">
        <s:element name="A" type="AType" minOccurs="0" maxOccurs="unbounded" />
        <s:element name="P" type="PType" minOccurs="0" maxOccurs="unbounded" />
      </s:choice>
    </s:sequence>
    <s:attribute name="ID" type="s:nonNegativeInteger" use="optional"/>
    <s:attribute name="F" use="optional">
      <s:simpleType>
        <s:restriction base="s:string">
          <s:pattern value="U?R?W?D?" />
```

210 / 224

```
      </s:restriction>
    </s:simpleType>
  </s:attribute>
  <s:attribute name="N" type="s:string" use="required"/>
  <s:attribute name="Y" type="s:string" use="optional"/>
  <s:attribute name="C" type="s:string" use="optional"/>

  <s:attribute name="R" type="s:string" use="optional"/>
  <s:attribute name="E" type="s:string" use="optional"/>
  <s:attribute name="Serial" type="s:string"  use="optional"/>
  <s:attribute name="T" type="s:string"  use="optional"/>
  <s:attribute name="Bound" use="optional">
    <s:simpleType>
      <s:restriction base="s:string">
        <s:enumeration value="True" />
      </s:restriction>
    </s:simpleType>
  </s:attribute>
  <s:attribute name="AssemblyQualifiedName" type="s:string" use="optional" />
</s:complexType>

<s:complexType name="RType">
  <s:sequence>
    <s:element name="Y" type="YTypeSimple" minOccurs="0" maxOccurs="unbounded" />
  </s:sequence>
  <s:attribute name="N" type="s:string" use="optional"/>
</s:complexType>

<s:complexType name="YType">
  <s:sequence>
    <s:choice maxOccurs="unbounded">
      <s:element name="P" type="PType" minOccurs="0" maxOccurs="unbounded" />
      <s:element name="A" type="AType" minOccurs="0" maxOccurs="unbounded" />
    </s:choice>
  </s:sequence>
  <s:attribute name="N" type="s:string" use="optional"/>
  <s:attribute name="ID" type="s:nonNegativeInteger" use="required"/>
  <s:attribute name="Base" type="s:nonNegativeInteger" use="optional"/>
  <s:attribute name="DataSource" use="optional">
    <s:simpleType>
      <s:restriction base="s:string">
        <s:enumeration value="True" />
      </s:restriction>
    </s:simpleType>
  </s:attribute>
  <s:attribute name="Designer" type="s:nonNegativeInteger" use="optional"/>
</s:complexType>

<s:complexType name="YTypeSimple">
  <s:sequence>
  </s:sequence>
  <s:attribute name="N" type="s:string" use="optional"/>
  <s:attribute name="ID" type="s:nonNegativeInteger" use="required"/>
</s:complexType>

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

211 / 224

```
            <s:choice maxOccurs="unbounded">
               <s:element name="Y" type="YType" minOccurs="0" maxOccurs="unbounded" />
               <s:element name="E" type="EType" minOccurs="0" maxOccurs="unbounded" />
            </s:choice>
         </s:sequence>
         <s:attribute name="N" type="s:string" use="required"/>
      </s:complexType>
   </s:element>

</s:schema>
```

AType: Specifies a .NET attribute of a class or property.

AType.Y: The number referring to the type of attribute to construct.

EType: Specifies an enumeration and the possible values of the enumeration.

EType.ID: The Identifier that other elements can use to refer to this enumeration type.

EType.Y: If the numerical type of the enumeration is not Int32, this number refers to the type of the enumeration.

EType.Value: Each **Value** element refers to one possible value for the enumeration. The text content of the element is the .NET numerical value of the enumeration value.

EType.Value.N: The .NET name of the enumeration value.

GType: Specifies an argument to an attribute constructor.

GType.Y: The number referring to the type of argument to pass to the attribute constructor.

GType.ID: If the value of **Y** is or exactly derives from **System.Type**, **ID** is the number of the type that is the value to be passed to the attribute constructor. If **ID** is set, **E**, **Serial**, **T**, and **Bound** is not set.

GType.E: If set, this specifies the value of the enumeration type to pass to the attribute constructor. If set, **T**, **Serial**, and **Bound** is not set.

GType.Serial: A base 64-encoded serialized form of the object to pass to the attribute constructor. If set, **T**, **E**, **Bound**, and **ID** is not set.

GType.T: A textual serialized object to pass to the attribute constructor. If set, **E**, **ID**, and **Serial** is not set.

PType: Specifies a property or nested property of a .NET type. This includes the default value of the property.

PType.Name: The name of the property.

PType.Y: The number that refers to the .NET type obtained or set by this property.

PType.Flags: Specifies the operations enabled for a property.

Character	Description
U	If the property cannot be used by operations supported by the client, the server returns this character as part of the **Flags** attribute.

Character	Description
R	If the property cannot be read, the server returns this character as part of the **Flags** attribute.
W	If the property can be written, the server returns this character as part of the **Flags** attribute.
D	If the property has a matching **System.Workflow.ComponentModel.DependencyProperty** object, the server returns this character as part of the **Flags** attribute.

PType.C: If a **System.ComponentModel.TypConverter** exists for the type of the property, the number referring to the type converter class is returned as this attribute.

PType.ID: If the value of **Y** is or derives from **System.Type**, this is the number of the type that is the default value of the property. If set, **E**, **Serial**, **T**, and **Bound** is not set.

PType.E: If set, this specifies the value of the enumeration type as the default value of the property. If set, **T**, **Serial**, and **Bound** is not set.

PType.Serial: A base 64-encoded serialized form of the object as the default value of the property. If set, **T**, **E**, **Bound**, and **ID** is not set.

PType.T: A textual serialized object as the default value of the property. If set, **E**, **ID**, and **Serial** is not set.

RType: This element refers to another assembly, and the **YTypeSimple** elements contained in it are the types from that external assembly that can be referenced later.

RType.N: The name of the assembly that contains the **YTypeSimple** elements.

YType: This element specifies a single type that exists with the assembly specified by the ancestor **Assembly** element.

YType.N: The name of the type.

YType.ID: The identifier for other elements to refer to this type in the ancestor assembly.

YType.Base: The number of the parent type of this **YType** that is part of the **baseTypes** list in the call to **GetAssemblyMetaData**.

YType.DataSource: If set to TRUE, this **YType** implements the **System.Web.UI.IDataSource** interface.

YType.Designer: If present, this is the number of the type used as the control designer for this **YType**.

YTypeSimple: This element matches a type from the parent **RType** element with a number that can be referenced by other elements to refer to this assembly-type pair.

YTypeSimple.N: The name of the type.

YTypeSimple.ID: The number available for other elements to refer to this assembly-type pair.

Assembly: This element contains all **RType**, **YType**, and **EType** objects that are present in the server system in the assembly named by the *assemblyName* parameter of the **GetAssemblyMetaData** method that are derived from or are required by sub elements of types

213 / 224

that derive from the set of base types specified in the *baseTypes* parameter of the **GetAssemblyMetaData** method.

Assembly.N: The name of the assembly.

<79> Section 3.1.4.9: Windows SharePoint Services 2.0 does not support this operation.

<80> Section 3.1.4.9: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 follow the ASP.NET resource localization model described in [MSDN-AWPRO].

<81> Section 3.1.4.9.2.1: In ASP.NET markup for binding to global resources, the first of two values specified is the **Class**, and the second value is the **ResourceID**. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this operation returns all resources of the specified **Class** (the value of the **ResourceName** parameter), regardless of **ResourceID**. The **Class** corresponds to the name (as described in [MSDN-AWPRO]) of a resX file on disk (in the format described in [MSDN-RIRF]), at wwwroot\wss\VirtualDirectories*80*\App_GlobalResources, where *80* is the port assigned to the server instance.

For example, the following ASP.NET Button control is bound to a localized resource as follows:

```
<asp:Button runat="server" Text="<%$ Resources:WSS, master_TurnOnAccessibility %>" />
```

The **Text** property of this control is bound to the localized resource with the class "WSS", and the **ResourceID** of "master_TurnOnAccessibility". This presumes that the server has a resource file named "WSS.resX" and any versions of this resource file at the correct location.

<82> Section 3.1.4.9.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the return value is XML-encoded XML markup in the resX file format described in [MSDN-RIRF].

<83> Section 3.1.4.10: Windows SharePoint Services 2.0 does not support this operation.

<84> Section 3.1.4.10.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, each **Assembly** element refers to one .NET assembly and namespace containing ASP.NET controls.

<85> Section 3.1.4.10.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Name** attribute contains the fully-qualified name of the assembly, in standard .NET assembly reference syntax. The assembly name is formatted in a way supported by an ASP.NET "@ Register" directive, as described in [MSDN-Register].

<86> Section 3.1.4.10.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **TagPrefix** attribute contains a suggested ASP.NET tag prefix to use for controls in this assembly and namespace. If present, the client uses this and the **Template** attribute of the **Control** element to generate markup for inserting that control. If this attribute is not present, the client programmatically generates a suitable tag prefix for this assembly and namespace. Windows SharePoint Services 3.0 and SharePoint Foundation 2010 use reflection on the control class and retrieve the value of **TagPrefixAttribute** if that attribute is present on the control. Otherwise they omit the attribute.

<87> Section 3.1.4.10.2.2: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 return all ASP.NET controls that meet all the following criteria:

▪ The assembly that contains the control is on the safe assembly list.

▪ The control has IsSafe and AllowRemoteDesigner set to TRUE.

▪ The control class derives from Control and not WebPart.

- The namespace that contains the control does not begin with "System".

- The control class has the ToolboxDataAttribute attribute applied.

<88> Section 3.1.4.10.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this is the class name without namespace or assembly qualification.

<89> Section 3.1.4.10.2.2: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 return the fully-qualified class name. This string is shown by the client in a tooltip when hovering over the control.

<90> Section 3.1.4.10.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, this attribute contains XML-encoded ASP.NET markup that the client uses when inserting this control. The string "{0}" is substituted for the control's tag prefix in the **Template** attribute. If this attribute is missing (this will not happen in the current implementation), the client instead generates appropriate ASP.NET markup by using the contents of the **Title** attribute.

<91> Section 3.1.4.11: Windows SharePoint Services 2.0 does not support this operation.

<92> Section 3.1.4.12: Windows SharePoint Services 3.0 does not support this operation.

<93> Section 3.1.4.12.2.2: In SharePoint Foundation 2010, the server reduces the **RowLimit** to 5 so that the client needs only five rows of data and expands any **AllFields** element so that the client has the list of all the **FieldRef** elements.

<94> Section 3.1.4.13: Windows SharePoint Services 2.0 does not support this operation.

<95> Section 3.1.4.14: Windows SharePoint Services 2.0 does not support this operation.

<96> Section 3.1.4.14: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the server returns a list of assemblies that contain all control elements that are marked as approved by server administrators as part of a configuration element.

<97> Section 3.1.4.16: Windows SharePoint Services 2.0 does not support this operation.

<98> Section 3.1.4.16.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the server can return both DWP and the format described in [MSDN-WPCDF].

<99> Section 3.1.4.18: Windows SharePoint Services 2.0 does not support this operation.

<100> Section 3.1.4.18: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the server returns markup similar to what is returned by the **get document** FPSE method defined in [MS-FPSE] section 3.1.5.3.6, but with additional annotations to the returned document, including but not limited to the following:

- If the source document on the server contained a Unicode **byte order mark**, it is removed from the document and replaced by the string "<HasByteOrderMark/>".

- ASP.NET controls in the page can have additional attributes applied to them that contain information to be used by the client application as rendering hints. These additional attributes have the namespace "__designer" (underscore underscore designer), such as "__designer:Preview", which contains HTML markup that can be used as a preview of what that ASP.NET control would render at runtime.

- <WebPart> tags are added to the markup of Web Parts on the page, containing Web Part properties in a format similar to that defined in [MSDN-WPCDF].

<101> Section 3.1.4.18.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the server returns markup similar to what is returned by the **get document** FPSE method defined in [MS-FPSE] section 3.1.5.3.6, but with additional annotations to the returned document, including but not limited to the following:

- If the source document on the server contained a Unicode byte order mark, it is removed from the document and replaced by the string "<HasByteOrderMark/>".

- ASP.NET controls in the page can have additional attributes applied to them that contain information to be used by the client application as rendering hints. These additional attributes have the namespace "__designer" (underscore underscore designer), such as "__designer:Preview", which contains HTML markup that can be used as a preview of what that ASP.NET control would render at runtime.

- <WebPart> tags are added to the markup of Web Parts on the page, containing Web Part properties in a format similar to that defined in [MSDN-WPCDF].

<102> Section 3.1.4.20: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, **GetWebPartPageDocument** returns similar, but not identical, results to **GetWebPartPage**. It is strongly recommended that implementers of new clients that operate with a Windows SharePoint Services 3.0 or a SharePoint Foundation 2010 server use **GetWebPartPage** instead.

<103> Section 3.1.4.20: The **GetWebPartPageDocument** operation is provided for compatibility with clients that interact with Windows SharePoint Services 2.0 servers and is obsolete in Windows SharePoint Services 3.0 and SharePoint Foundation 2010. Windows SharePoint Services 3.0 and SharePoint Foundation 2010 do not support this operation on pages that contain certain new constructs, such as ASP.NET Web Parts.

<104> Section 3.1.4.20.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the server returns markup similar to what is returned by the **get document** FPSE method defined in [MS-FPSE] section 3.1.5.3.6, but with additional annotations to the returned document, including but not limited to the following:

- If the source document on the server contained a Unicode byte order mark, it is removed from the document and replaced by the string "<HasByteOrderMark/>".

- ASP.NET controls in the page can have additional attributes applied to them that contain information to be used by the client application as rendering hints. These additional attributes have the namespace "__designer" (underscore underscore designer), such as "__designer:Preview", which contains HTML markup that can be used as a preview of what that ASP.NET control would render at runtime.

- <WebPart> tags are added to the markup of Web Parts on the page, containing Web Part properties in a format similar to that defined in [MSDN-WPCDF].

<105> Section 3.1.4.22: Windows SharePoint Services 2.0 does not support this operation.

<106> Section 3.1.4.22.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the supported formats for this operation are the DWP format defined in **wpv2:WebPart** (section 2.2.4.2) or the .webpart format defined in webParts (section 2.2.3.1).

<107> Section 3.1.4.23.2.1: An example of a query supported in Windows SharePoint Services 3.0 and SharePoint Foundation 2010 can be found at [MSDN-GETXML].

<108> Section 3.1.4.24: Windows SharePoint Services 2.0 does not support this operation.

<109> Section 3.1.4.25.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Properties** element contains XML-encoded content in the DWP format defined in **wpv2:WebPart** (section 2.2.3.2).

<110> Section 3.1.4.25.2.2: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **Results.Result.DesignerInfo.Preview** contains HTML.

<111> Section 3.1.4.26.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the formats supported for the **webPartXml** property are the DWP format defined in **wpv2:WebPart** (section 2.2.4.2) and the .webpart format defined in webParts (section 2.2.3.1).

<112> Section 3.1.4.27: Windows SharePoint Services 2.0 does not support this operation.

<113> Section 3.1.4.27.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the formats supported for the **webPartXml** property are DWP format defined in **wpv2:WebPart** (section 2.2.4.2) and the .webpart format defined in webParts (section 2.2.3.1).

<114> Section 3.1.4.28: Windows SharePoint Services 2.0 does not support this operation.

<115> Section 3.1.4.28.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the workflowMarkupText element contains XML-encoded XAML markup, as described in [MSDN-UWM].

<116> Section 3.1.4.28.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **rulesText** element contains XML-encoded RULES markup, as described in [MSDN-UWM].

<117> Section 3.1.4.28.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **configBlob** element contains XML-encoded markup in the format of a workflow configuration file, as described in [MSDN-WCSO]. If the value of the flag parameter is "1", this element is empty.

The following is the schema for the workflow configuration markup:

```
<s:schema elementFormDefault="qualified">
   <s:element name="WorkflowConfig">
      <s:complexType>
         <s:sequence>
            <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
            <s:element name="Association" minOccurs="0" maxOccurs="1">
               <s:complexType>
                  <s:sequence>
                  </s:sequence>
                  <s:attribute name="TaskListID" type="s:string" use="optional" />
                  <s:anyAttribute />
               </s:complexType>
            </s:element>
            <s:element name="ContentTypes" minOccurs="0" maxOccurs="1">
               <s:complexType>
               <s:sequence>
                  <s:element name="ContentType" minOccurs="0" maxOccurs="unbounded">
                     <s:complexType>
                        <s:sequence>
                           <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
                        </s:sequence>
                        <s:attribute name="ContentTypeID" type="s:string" use="required"
/>
                        <s:attribute name="Form" type="s:string" />
                        <s:anyAttribute />
                     </s:complexType>
```

```
                    </s:element>
                </s:sequence>
            </s:complexType>
        </s:element>
        <s:any namespace="##any" minOccurs="0" maxOccurs="unbounded" />
    </s:sequence>
        </s:complexType>
    </s:element>
</s:schema>
```

WorkflowConfig: The root element of the workflow configuration file.

WorkflowConfig.Association: Information about how the client authoring the workflow can later associate the workflow with objects on the server.

WorkflowConfig.Association.TaskListID: The GUID of a list (1) that will have different content types applied to it. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if this attribute is empty or null, the server selects an existing task list (1) or creates a new task list (1) with a default name. If the server either reuses an existing list (1) or creates a new task list (1), all future references to **WorkflowConfig.Association.TaskListID** use the value of the GUID of the list (1).

WorkflowConfig.Association.HistoryListID: The GUID of a list (1) in which to place list items that record information about the workflow. In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, if this attribute is empty or null, the server selects an existing list (1) or creates a new list (1) with a default name. If the server either reuses an existing list (1) or creates a new list (1), all future references to **WorkflowConfig.Association.HistoryListID** use the value of the GUID of the list (1).

WorkflowConfig.ContentTypes: The container for all **ContentType** elements that the client requests be present on the task list (1) identified by **WorkflowConfig**.**Association**.**TaskListID**.

WorkflowConfig.ContentTypes.ContentType: Specifies one content type requested by the client to be present on the task list (1) identified by **WorkflowConfig**.**Association**.**TaskListID**.

WorkflowConfig.ContentTypes.ContentType.ContentTypeID: The identifier of the content type in the form of a hexadecimal number.

WorkflowConfig.ContentTypes.ContentType.Form: The site-relative URL of the form to be used when editing an item of the **ContentTypeID** content type in the **WorkflowConfig.Association.TaskListID** list (1).

WorkflowConfig.ContentTypes.ContentType.WorkflowForm: If preceded by a slash (/), this string is a **site-collection relative URL**. If not, it is a site-relative URL. This specifies the URL of the form to be used when editing an item of the **ContentTypeID** content type on the **WorkflowConfig.Association.TaskListID** list (1). If **Form** is set, the server ignore this value. This attribute is ignored by Windows SharePoint Services 3.0.

<118> Section 3.1.4.28.2.1: In Windows SharePoint Services 3.0 and SharePoint Foundation 2010, the **flag** parameter is used to control whether the server creates support objects for the workflow. The following table summarizes the values for **flag** supported by Windows SharePoint Services 3.0 and SharePoint Foundation 2010.

Flag	Description
1	The server validates **workflowMarkupText** and **rulesText**, if present, and the server ignore

218 / 224

Flag	Description
	configBlob. In this mode, the server merely validates the content provided and does not make permanent configuration changes as a result.
2	The server validates **workflowMarkupText** and **rulesText**, if present, and **configBlob**. In this mode, the server validates the content provided and creates the support objects for this workflow.

The following table summarizes whether each element is required to have content, based on the value of the **flag** element.

Flag	workflowMarkupText	rulesText	configBlob
1	Required non-empty.	Optionally empty.	Ignored
2	Required non-empty.	Optionally empty.	Required

<119> Section 3.1.4.28.2.2: Windows SharePoint Services 3.0 and SharePoint Foundation 2010 return the GUID of the task list (1) for the workflow.

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: September 12, 2012