

# [MS-SHDACCWS]: Shared Access 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](mailto: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

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
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Editorial	Revised and edited the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Minor	Clarified the meaning of the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	Editorial	Changed language and formatting in the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
<b>2 Messages</b>	<b>8</b>
2.1 Transport	8
2.2 Common Message Syntax	8
2.2.1 Namespaces	8
2.2.2 Messages	8
2.2.3 Elements	9
2.2.4 Complex Types	9
2.2.5 Simple Types	9
2.2.6 Attributes	9
2.2.7 Groups	9
2.2.8 Attribute Groups	9
2.2.9 Common Data Structures	9
<b>3 Protocol Details</b>	<b>10</b>
3.1 Server Details	10
3.1.1 Abstract Data Model	10
3.1.2 Timers	10
3.1.3 Initialization	10
3.1.4 Message Processing Events and Sequencing Rules	10
3.1.4.1 IsOnlyClient	10
3.1.4.1.1 Messages	11
3.1.4.1.1.1 IsOnlyClientSoapIn	11
3.1.4.1.1.2 IsOnlyClientSoapOut	11
3.1.4.1.2 Elements	11
3.1.4.1.2.1 IsOnlyClient	11
3.1.4.1.2.2 IsOnlyClientResponse	12
3.1.5 Timer Events	12
3.1.6 Other Local Events	12
<b>4 Protocol Examples</b>	<b>13</b>
4.1 IsOnlyClient	13
<b>5 Security</b>	<b>14</b>
5.1 Security Considerations for Implementers	14
5.2 Index of Security Parameters	14
<b>6 Appendix A: Full WSDL</b>	<b>15</b>

<b>7 Appendix B: Product Behavior .....</b>	<b>17</b>
<b>8 Change Tracking.....</b>	<b>18</b>
<b>9 Index .....</b>	<b>20</b>

Preliminary

# 1 Introduction

This document specifies the Shared Access Web Service Protocol that is used for determining whether a document is being co-authored.

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

## 1.1 Glossary

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

**Hypertext Transfer Protocol (HTTP)**  
**Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**

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

**co-authoring transition**  
**Simple Object Access Protocol (SOAP)**  
**site**  
**SOAP action**  
**SOAP body**  
**SOAP envelope**  
**SOAP fault**  
**Uniform Resource Locator (URL)**  
**Web Services Description Language (WSDL)**  
**WSDL operation**  
**XML namespace**  
**XML namespace prefix**  
**XML schema**

The following terms are specific to this document:

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

## 1.2 References

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

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto: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-LISTSWS] Microsoft Corporation, "[Lists Web Service 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>

[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

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

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

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

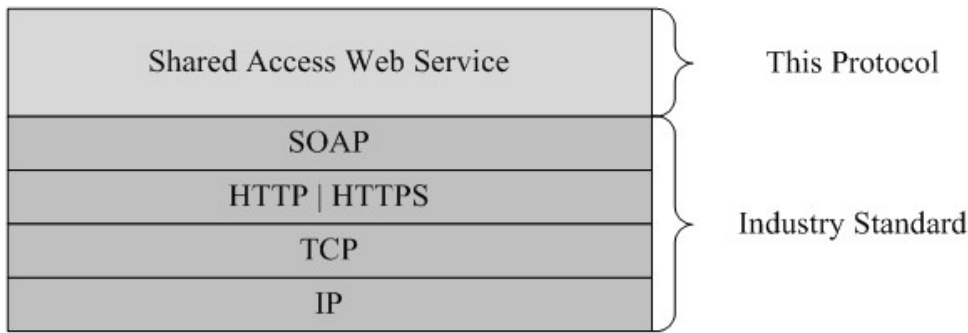
### 1.3 Protocol Overview (Synopsis)

The protocol allows clients to determine whether a **co-authoring transition** request was made for a document. A typical scenario for using this protocol is an authoring client that allows multiple users edit a document in a co-authoring session.

### 1.4 Relationship to Other Protocols

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

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



**Figure 1: This protocol in relation to other protocols**

## 1.5 Prerequisites/Preconditions

This protocol operates against a **site(1)** that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending `"/_vti_bin/sharedaccess.asmx"` to the URL of the site(1), for example `http://www.contoso.com/Repository/_vti_bin/sharedaccess.asmx`.

## 1.6 Applicability Statement

This protocol can be used by a protocol client to determine if it is the only client currently editing a document stored on a collaboration server, or alternately, if it should transition to a shared editing mode.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

**Supported Transports:** This protocol uses multiple transports with Simple Object Access Protocol (SOAP) as described in section [2.1](#).

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.

## 2 Messages

### 2.1 Transport

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

Protocol messages MUST be formatted as specified either in [\[SOAP1.1\]](#), Section 4, **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. or in [\[SOAP1.2/1\]](#), Section 5.4.

### 2.2 Common Message Syntax

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

#### 2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
soap	<a href="http://schemas.xmlsoap.org/wsd/soap/">http://schemas.xmlsoap.org/wsd/soap/</a>	<a href="#">[SOAP1.1]</a>
tns	<a href="http://schemas.microsoft.com/sharepoint/soap/">http://schemas.microsoft.com/sharepoint/soap/</a>	<a href="#">[MS-LISTSWS]</a>
s1	<a href="http://microsoft.com/wsd/types/">http://microsoft.com/wsd/types/</a>	[MS-LISTSWS]
s	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a>
soap12	<a href="http://schemas.xmlsoap.org/wsd/soap12/">http://schemas.xmlsoap.org/wsd/soap12/</a>	<a href="#">[SOAP1.2/1]</a> <a href="#">[SOAP1.2/2]</a>
wsd	<a href="http://schemas.xmlsoap.org/wsd/">http://schemas.xmlsoap.org/wsd/</a>	<a href="#">[WSDL]</a>
mime	<a href="http://schemas.xmlsoap.org/wsd/mime/">http://schemas.xmlsoap.org/wsd/mime/</a>	
http	<a href="http://schemas.xmlsoap.org/wsd/http/">http://schemas.xmlsoap.org/wsd/http/</a>	
tm	<a href="http://microsoft.com/wsd/mime/textMatching/">http://microsoft.com/wsd/mime/textMatching/</a>	
soapenc	<a href="http://schemas.xmlsoap.org/soap/encoding/">http://schemas.xmlsoap.org/soap/encoding/</a>	
core	<a href="http://schemas.microsoft.com/sharepoint/soap/">http://schemas.microsoft.com/sharepoint/soap/</a>	<a href="#">[MS-WSSCAML]</a>

#### 2.2.2 Messages

This specification does not define any common WSDL message definitions.



### **2.2.3 Elements**

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

### **2.2.4 Complex Types**

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

### **2.2.5 Simple Types**

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

### **2.2.6 Attributes**

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

### **2.2.7 Groups**

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

### **2.2.8 Attribute Groups**

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

### **2.2.9 Common Data Structures**

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

## 3 Protocol Details

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

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

### 3.1 Server Details

All operations consist of a basic request-response pair and the server treats each request as an independent transaction that is unrelated to any previous request.

#### 3.1.1 Abstract Data Model

None.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
<b>IsOnlyClient</b>	Specifies whether a co-authoring transition request was made for a document.

##### 3.1.4.1 IsOnlyClient

The method is used by a protocol client to determine whether a co-authoring transition request was made for a document.

```
<wsdl:operation name="IsOnlyClient">
  <wsdl:input message="tns:IsOnlyClientSoapIn" />
  <wsdl:output message="tns:IsOnlyClientSoapOut" />
</wsdl:operation>
```

The protocol client sends an **IsOnlyClientSoapIn** request message and the protocol server responds with an **IsOnlyClientSoapOut** 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
<b>IsOnlyClientSoapIn</b>	The request to determine whether a co-authoring transition request was made for a document.
<b>IsOnlyClientSoapOut</b>	The response to a request to determine whether a co-authoring transition request was made for a document.

#### 3.1.4.1.1.1 IsOnlyClientSoapIn

This message is the request of the **IsOnlyClient** operation.

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

```
http://schemas.microsoft.com/sharepoint/soap/IsOnlyClient
```

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

#### 3.1.4.1.1.2 IsOnlyClientSoapOut

This message is the response of the **IsOnlyClient** operation.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/IsOnlyClient
```

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

### 3.1.4.1.2 Elements

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

Element	Description
<b>IsOnlyClient</b>	The request to determine whether a co-authoring transition request was made for a document.
<b>IsOnlyClientResponse</b>	Contains the response to a request to determine whether a co-authoring transition request was made for a document.

#### 3.1.4.1.2.1 IsOnlyClient

The **IsOnlyClient** element defines the input parameters for **IsOnlyClient** operation.

```
<s:element name="IsOnlyClient">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="id"
type="core:UniqueIdentifierWithOrWithoutBraces" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**id** : The identifier of the document in the server. Note that core:UniqueIdentifierWithOrWithoutBraces is specified in [\[MS-WSSCAML\]](#) section 2.1.18.

#### 3.1.4.1.2.2 IsOnlyClientResponse

IsOnlyClientResponse specifies the output of the **IsOnlyClient** operation.

```
<s:element name="IsOnlyClientResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="1" maxOccurs="1" name="IsOnlyClientResult" type="s:boolean" />
    </s:sequence>
  </s:complexType>
</s:element>
```

**IsOnlyClientResult** : The value of this element **MUST** be false if there was a co-authoring transition request for the document. In all other cases, the value **MUST** be true.

#### 3.1.5 Timer Events

None.

#### 3.1.6 Other Local Events

None.

## 4 Protocol Examples

### 4.1 IsOnlyClient

Overall scenario: A protocol client wants to know if a document is transitioning into co-authoring mode. It sends a request to the server to verify if it is the only client editing the document.

The following example shows a sample request where the id element refers to the document identifier.

```
<?xml version="1.0" encoding="utf-8"?>
<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>
    <IsOnlyClient xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <id>{3F2504E0-4F89-11D3-9A0C-0305E82C3301}</id>
    </IsOnlyClient>
  </soap:Body>
</soap:Envelope>
```

The following example shows a sample response from the server.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <IsOnlyClientResponse xmlns="http://schemas.microsoft.com/sharepoint/soap/">
      <IsOnlyClientResult>true</IsOnlyClientResult>
    </IsOnlyClientResponse>
  </soap:Body>
</soap:Envelope>
```

## 5 Security

### 5.1 Security Considerations for Implementers

This protocol introduces no additional security considerations beyond those applicable to its underlying protocols.

### 5.2 Index of Security Parameters

None.

Preliminary

## 6 Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```
<?xml version="1.0"?>
<wSDL:definitions xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
xmlns:tm="http://microsoft.com/wSDL/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wSDL/mime/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/"
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://schemas.microsoft.com/sharepoint/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:core="http://schemas.microsoft.com/sharepoint/soap/"
  <wSDL:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/"
  <s:import namespace="http://microsoft.com/wSDL/types/" />
    <s:element name="IsOnlyClient">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="id"
type="core:UniqueIdentifierWithOrWithoutBraces" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="IsOnlyClientResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="IsOnlyClientResult" type="s:boolean"
/>
        </s:sequence>
      </s:complexType>
    </s:element>
  </s:schema>
  <s:schema elementFormDefault="qualified"
targetNamespace="http://microsoft.com/wSDL/types/"
  </s:schema>
</wSDL:types>
<wSDL:message name="IsOnlyClientSoapIn">
  <wSDL:part name="parameters" element="tns:IsOnlyClient" />
</wSDL:message>
<wSDL:message name="IsOnlyClientSoapOut">
  <wSDL:part name="parameters" element="tns:IsOnlyClientResponse" />
</wSDL:message>
<wSDL:portType name="SharedAccessSoap">
  <wSDL:operation name="IsOnlyClient">
    <wSDL:input message="tns:IsOnlyClientSoapIn" />
    <wSDL:output message="tns:IsOnlyClientSoapOut" />
  </wSDL:operation>
</wSDL:portType>
<wSDL:binding name="SharedAccessSoap" type="tns:SharedAccessSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wSDL:operation name="IsOnlyClient">
    <soap:operation soapAction="http://schemas.microsoft.com/sharepoint/soap/IsOnlyClient"
style="document" />
  <wSDL:input>
    <soap:body use="literal" />
  </wSDL:input>
</wSDL:binding>
</wSDL:definitions>
```

```
</wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="SharedAccessSoap12" type="tns:SharedAccessSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="IsOnlyClient">
    <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/IsOnlyClient" 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® SharePoint® Foundation 2010
- Microsoft® Office 15 Technical Preview
- Microsoft® SharePoint® Foundation 15 Technical 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.

## 8 Change Tracking

This section identifies changes that were made to the [MS-SHDACCWS] protocol document between the June 2011 and January 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

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

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

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

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

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

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

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

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

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

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

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">3.1.4.1.1 Messages</a>	Added a table with summary descriptions of the messages.	N	New content added for template compliance.
<a href="#">3.1.4.1.2 Elements</a>	Added a table with summary descriptions of the elements.	N	New content added for template compliance.
<a href="#">Z Appendix B: Product Behavior</a>	Updated the product list.	Y	Content updated.

## 9 Index

### A

Abstract data model  
[server](#) 10  
[Applicability](#) 7  
[Attribute groups](#) 9  
[Attributes](#) 9

### C

[Capability negotiation](#) 7  
[Change tracking](#) 18  
Client  
[overview](#) 10  
[Common data structures](#) 9  
[Complex types](#) 9

### D

Data model - abstract  
[server](#) 10  
Details  
[server](#) 10

### E

Events  
[local - server](#) 12  
[timer - server](#) 12

Examples  
[IsOnlyClient](#) 13

### F

[Fields - vendor-extensible](#) 7  
[Full WSDL](#) 15

### G

[Glossary](#) 5  
[Groups](#) 9

### I

[Implementer - security considerations](#) 14  
[Index of security parameters](#) 14  
[Informative references](#) 6  
Initialization  
[server](#) 10  
[Introduction](#) 5  
[IsOnlyClient example](#) 13

### L

Local events  
[server](#) 12

### M

Message processing  
[server](#) 10  
Messages  
[attribute groups](#) 9  
[attributes](#) 9  
[common data structures](#) 9  
[complex types](#) 9  
[elements](#) 9  
[enumerated](#) 8  
[groups](#) 9  
[namespaces](#) 8  
[simple types](#) 9  
[syntax](#) 8  
[transport](#) 8

### N

[Namespaces](#) 8  
[Normative references](#) 5

### O

Operations  
[IsOnlyClient](#) 10  
[Overview \(synopsis\)](#) 6

### P

[Parameters - security index](#) 14  
[Preconditions](#) 7  
[Prerequisites](#) 7  
[Product behavior](#) 17

### R

[References](#) 5  
[informative](#) 6  
[normative](#) 5  
[Relationship to other protocols](#) 6

### S

Security  
[implementer considerations](#) 14  
[parameter index](#) 14  
Sequencing rules  
[server](#) 10  
Server  
[abstract data model](#) 10  
[initialization](#) 10  
[IsOnlyClient operation](#) 10  
[local events](#) 12  
[message processing](#) 10  
[overview](#) 10  
[sequencing rules](#) 10  
[timer events](#) 12  
[timers](#) 10  
[Server details](#) 10

[Simple types](#) 9  
[Standards assignments](#) 7  
Syntax  
  [messages - overview](#) 8

## T

Timer events  
  [server](#) 12  
Timers  
  [server](#) 10  
[Tracking changes](#) 18  
[Transport](#) 8  
Types  
  [complex](#) 9  
  [simple](#) 9

## V

[Vendor-extensible fields](#) 7  
[Versioning](#) 7

## W

[WSDL](#) 15