

[MS-OBPAS]: Office Broadcast Participant Service 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.

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References.....	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Overview	7
1.4	Relationship to Other Protocols.....	8
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields.....	8
1.9	Standards Assignments	9
2	Messages.....	10
2.1	Transport.....	10
2.2	Common Message Syntax	10
2.2.1	Namespaces	10
2.2.2	Messages	11
2.2.3	Elements.....	11
2.2.4	Complex Types	11
2.2.4.1	ArrayOfKeyValueOfstringstring	12
2.2.4.2	ArrayOfPPTBroadcastAnimationStepData	12
2.2.4.3	ArrayOfPPTBroadcastMediaStateData	13
2.2.4.4	EnumTypes	13
2.2.4.5	PPTBroadcastAnimationStepData	13
2.2.4.6	PPTBroadcastMediaStateData	14
2.2.4.7	PPTStateData	14
2.2.4.8	ServiceError.....	15
2.2.4.9	ServiceResult	16
2.2.4.10	WordStateData	16
2.2.5	Simple Types	16
2.2.5.1	AppType	17
2.2.5.2	BroadcastAppCapability	18
2.2.5.3	BroadcastDataKey	18
2.2.5.4	BroadcastState.....	19
2.2.5.5	char.....	20
2.2.5.6	ClientActions	20
2.2.5.7	duration	21
2.2.5.8	guid.....	21
2.2.5.9	PPTMediaPlaybackState	21
2.2.5.10	PPTSlideShowState	22
2.2.5.11	ServiceErrorType	22
2.2.6	Attributes.....	23
2.2.7	Groups.....	23
2.2.8	Attribute Groups	23
3	Protocol Details.....	24
3.1	Server Details	24
3.1.1	Abstract Data Model	25
3.1.2	Timers	25

3.1.3	Initialization	25
3.1.4	Message Processing Events and Sequencing Rules	25
3.1.4.1	BroadcastGetData	26
3.1.4.1.1	Messages	26
3.1.4.1.1.1	IParticipantService_BroadcastGetData_InputMessage	26
3.1.4.1.1.2	IParticipantService_BroadcastGetData_OutputMessage	26
3.1.4.1.2	Elements	26
3.1.4.1.2.1	BroadcastGetData	27
3.1.4.1.2.2	BroadcastGetDataResponse	27
3.1.4.1.3	Complex Types	28
3.1.4.1.3.1	BroadcastUser	28
3.1.4.1.4	Simple Types	29
3.1.4.1.5	Attributes	29
3.1.4.1.6	Groups	29
3.1.4.1.7	Attribute Groups	29
3.1.4.2	BroadcastJoinSession	29
3.1.4.2.1	Messages	29
3.1.4.2.1.1	IParticipantService_BroadcastJoinSession_InputMessage	30
3.1.4.2.1.2	IParticipantService_BroadcastJoinSession_OutputMessage	30
3.1.4.2.2	Elements	30
3.1.4.2.2.1	BroadcastJoinSession	30
3.1.4.2.2.2	BroadcastJoinSessionResponse	31
3.1.4.2.3	Complex Types	31
3.1.4.2.4	Simple Types	31
3.1.4.2.5	Attributes	31
3.1.4.2.6	Groups	31
3.1.4.2.7	Attribute Groups	31
3.1.4.3	BroadcastUnjoinSession	31
3.1.4.3.1	Messages	32
3.1.4.3.1.1	IParticipantService_BroadcastUnjoinSession_InputMessage	32
3.1.4.3.1.2	IParticipantService_BroadcastUnjoinSession_OutputMessage	32
3.1.4.3.2	Elements	32
3.1.4.3.2.1	BroadcastUnjoinSession	33
3.1.4.3.2.2	BroadcastUnjoinSessionResponse	33
3.1.4.3.3	Complex Types	33
3.1.4.3.4	Simple Types	33
3.1.4.3.5	Attributes	33
3.1.4.3.6	Groups	33
3.1.4.3.7	Attribute Groups	34
3.1.4.4	BroadcastPing	34
3.1.4.4.1	Messages	34
3.1.4.4.1.1	IParticipantService_BroadcastPing_InputMessage	34
3.1.4.4.1.2	IParticipantService_BroadcastPing_OutputMessage	34
3.1.4.4.2	Elements	35
3.1.4.4.2.1	BroadcastPing	35
3.1.4.4.2.2	BroadcastPingResponse	35
3.1.4.4.3	Complex Types	35
3.1.4.4.4	Simple Types	35
3.1.4.4.5	Attributes	35
3.1.4.4.6	Groups	35
3.1.4.4.7	Attribute Groups	36
3.1.5	Timer Events	36
3.1.6	Other Local Events	36

4 Protocol Examples	37
4.1 Attendee Client Example	37
5 Security	40
5.1 Security Considerations for Implementers	40
5.2 Index of Security Parameters	40
6 Appendix A: Full WSDL	41
7 Appendix B: Full XML Schema	44
7.1	
http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data Schema	44
7.2	
http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface Schema	46
7.3 http://schemas.microsoft.com/2003/10/Serialization/Arrays Schema	47
7.4 http://schemas.microsoft.com/2003/10/Serialization/ Schema	48
7.5	
http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/ Schema	49
8 Appendix C: Product Behavior	51
9 Change Tracking	52
10 Index	54

1 Introduction

The Office Broadcast Participant Service Protocol enables a protocol client to retrieve information about the state of a document being shared on a protocol server.

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 [\[MS-GLOS\]](#):

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

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

broadcast session
presentation slide
Simple Object Access Protocol (SOAP)
slide show
SOAP action
SOAP body
SOAP fault
time code
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
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 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-OBPRS] Microsoft Corporation, "[Office Broadcast Presentation Service Specification](#)".

[MS-WOPI] Microsoft Corporation, "[Web Application Open Platform Interface Protocol 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>

[RFC4627] Crockford, D., "The application/json Media Type for Javascript Object Notation (JSON)", RFC 4627, July 2006, <http://www.ietf.org/rfc/rfc4627.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>

[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>

[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>

1.3 Overview

When meeting remotely, it is useful to share the current view of a document so others can follow along with the document sharer. To share this view in real-time and allow attendees to follow along with the presenter as they move through the document, information regarding the presenter's current view of the document needs to be shared between the presenter and attendees.

This protocol enables a protocol client to send requests to a protocol server allowing the client to join an in-progress document **broadcast session**, and to retrieve data about the state of a broadcast session on the protocol server.

1.4 Relationship to Other Protocols

This protocol uses the **Simple Object Access Protocol (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 the **Hypertext Transfer Protocol (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:

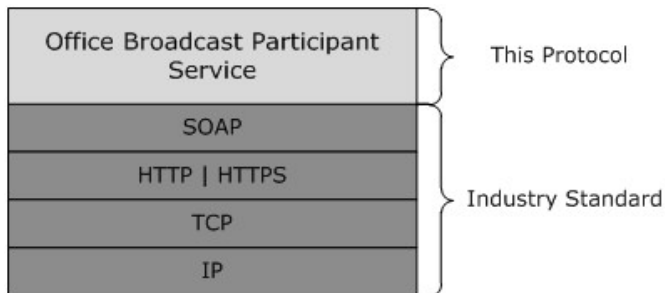


Figure 1: This protocol in relation to other protocols

This protocol works in conjunction with the Office Broadcast Presentation Service protocol as described in [\[MS-OBPRS\]](#) and uses the Web Application Open Platform Interface protocol for file retrieval as described in [\[MS-WOPI\]](#).

1.5 Prerequisites/Preconditions

This protocol operates between a protocol client and a protocol server that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending "/m/met/Participant.svc" to the URL of the site, for example:
`http://www.example.com/m/met/Participant.svc`

This protocol requires that a broadcast session be started and in progress prior to joining it. Broadcast sessions can be started using the Office Broadcast Presentation Service Protocol as defined in [\[MS-OBPRS\]](#).

This protocol requires that files being used in broadcast sessions be identified and accessed as specified in [\[MS-WOPI\]](#).

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

1.6 Applicability Statement

This protocol is designed to join in-progress broadcast sessions on the protocol server and to retrieve broadcast session information on the protocol server.

1.7 Versioning and Capability Negotiation

This protocol uses multiple transports with SOAP, as specified in section [2.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

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

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 or in [\[SOAP1.2/1\]](#) section 5. Protocol server faults **MUST** be returned either using HTTP Status Codes, as specified in [\[RFC2616\]](#) section 10 or using **SOAP faults**, as specified in either [\[SOAP1.1\]](#) 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 specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific **XML namespace prefix** for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefi x	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data	
tns1	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface	
tns2	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
tns3	http://schemas.microsoft.com/2003/10/Serialization/	
tns4	http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/	
tns5	http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Imports	
wsa w	http://www.w3.org/2006/05/addressing/wsdl	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

Prefi x	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEM A1] [XMLSCHEM A2]

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

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
ArrayOfKeyValueOfstringstring	A complex type that specifies a list of key/value pairs.
ArrayOfPPTBroadcastAnimationStepData	A complex type that specifies a list of PPTBroadcastAnimationStepData (section 2.2.4.5) elements. Each element in the list specifies a step in an animation timeline. Each entry in the list MUST have a unique value in its TimelineId (section 2.2.4.5) field.
ArrayOfPPTBroadcastMediaStateData	A complex type that specifies a list of PPTBroadcastMediaStateData (section 2.2.4.6) elements. Each element in the list specifies a state for a multimedia object. Each entry in the list MUST have a unique value in its MediaId field.
EnumTypes	This complex type is reserved and MUST be ignored.
PPTBroadcastAnimationStepData	A complex type that specifies a step in an animation timeline.
PPTBroadcastMediaStateData	A complex type that specifies a state for a multimedia object such as an embedded video clip.
PPTStateData	A complex type that specifies data about the state of a broadcast session of a presentation.
ServiceError	A complex type that specifies error information returned from the protocol server to a protocol client.
ServiceResult	A complex type that specifies the result of a protocol method. The protocol server returns this type to the protocol client containing either a successful Result element or an Error element.
WordStateData	A complex type that specifies data about the state of a

Complex type	Description
	broadcast session of a document.

2.2.4.1 ArrayOfKeyValueOfstringstring

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

A complex type that specifies a list of key/value pairs.

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

KeyValueOfstringstring: A key/value pair. This element **MUST** be present.

KeyValueOfstringstring.Key: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the key in a key/value pair. This element **MUST** be present.

KeyValueOfstringstring.Value: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the value in a key/value pair. This element **MUST** be present.

2.2.4.2 ArrayOfPPTBroadcastAnimationStepData

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A complex type that specifies a list of **PPTBroadcastAnimationStepData** (section [2.2.4.5](#)) elements. Each element in the list specifies a step in an animation timeline. Each entry in the list **MUST** have a unique value in its **TimelineId** (section [2.2.4.5](#)) field.

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

PPTBroadcastAnimationStepData: Specifies a **PPTBroadcastAnimationStepData** (section [2.2.4.5](#)) element. The element **MUST** be present.

2.2.4.3 ArrayOfPPTBroadcastMediaStateData

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A complex type that specifies a list of **PPTBroadcastMediaStateData** (section [2.2.4.6](#)) elements. Each element in the list specifies a state for a multimedia object. Each entry in the list MUST have a unique value in its **MediaId** field.

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

PPTBroadcastMediaStateData: Specifies a **PPTBroadcastMediaStateData** element. This element MUST be present.

2.2.4.4 EnumTypes

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

This complex type is reserved and MUST be ignored.

```
<xs:complexType name="EnumTypes" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="AppType" type="tns:AppType"/>
    <xs:element minOccurs="0" name="BroadcastAppCapability"
      type="tns:BroadcastAppCapability"/>
    <xs:element minOccurs="0" name="BroadcastDataKey" type="tns:BroadcastDataKey"/>
    <xs:element minOccurs="0" name="BroadcastState" type="tns:BroadcastState"/>
    <xs:element minOccurs="0" name="PPTMediaPlaybackState" type="tns:PPTMediaPlaybackState"/>
    <xs:element minOccurs="0" name="PPTSlideShowState" type="tns:PPTSlideShowState"/>
  </xs:sequence>
</xs:complexType>
```

AppType: This element is reserved and MUST be ignored.

BroadcastAppCapability: This element is reserved and MUST be ignored.

BroadcastDataKey: This element is reserved and MUST be ignored.

BroadcastState: This element is reserved and MUST be ignored.

PPTMediaPlaybackState: This element is reserved and MUST be ignored.

PPTSlideShowState: This element is reserved and MUST be ignored.

2.2.4.5 PPTBroadcastAnimationStepData

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A complex type that specifies a step in an animation timeline.

```
<xs:complexType name="PPTBroadcastAnimationStepData"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Step" type="xs:int"/>
    <xs:element minOccurs="0" name="TimelineId" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Step: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) element that specifies the step number in the animation timeline given by the **TimelineId** field. This element **MUST** be present.

TimelineId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies an identifier of the animation timeline. This element **MUST** be present.

2.2.4.6 PPTBroadcastMediaStateData

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data>

A complex type that specifies a state for a multimedia object such as an embedded video clip.

```
<xs:complexType name="PPTBroadcastMediaStateData"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="AtPosition" type="xs:double"/>
    <xs:element minOccurs="0" name="MediaId" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="State" type="tns:PPTMediaPlaybackState"/>
  </xs:sequence>
</xs:complexType>
```

AtPosition: An **xs:double** ([\[XMLSCHEMA2\]](#) section 3.2.5) element that specifies the **time code** of the multimedia object identified by the **MediaId** field when the **State** field is changed. This element **MUST** be present.

MediaId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies an identifier of the multimedia object. This element **MUST** be present.

State: A **PPTMediaPlaybackState** (section [2.2.5.9](#)) element that specifies the playback state of the multimedia object identified by the **MediaId** field. This element **MUST** be present.

2.2.4.7 PPTStateData

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data>

A complex type that specifies data about the state of a broadcast session of a presentation.

```
<xs:complexType name="PPTStateData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="AnimationStepDataList" nillable="true"
      type="tns:ArrayOfPPTBroadcastAnimationStepData"/>
    <xs:element minOccurs="0" name="MediaStateDataList" nillable="true"
      type="tns:ArrayOfPPTBroadcastMediaStateData"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:element minOccurs="0" name="PPTSlideShowState" type="tns:PPTSlideShowState"/>
<xs:element minOccurs="0" name="SlideId" type="xs:unsignedInt"/>
<xs:element minOccurs="0" name="SlideIndex" type="xs:unsignedInt"/>
</xs:sequence>
</xs:complexType>

```

AnimationStepDataList: An **ArrayOfPPTBroadcastAnimationStepData** (section [2.2.4.2](#)) element that specifies the state of each of the animation timelines. This element **MUST** be present.

MediaStateDataList: An **ArrayOfPPTBroadcastMediaStateData** (section [2.2.4.3](#)) element that specifies the state of each of the multimedia objects. This element **MUST** be present.

PPTSlideShowState: A **PPTSlideShowState** (section [2.2.5.10](#)) element that specifies the current state of the **slide show**. This element **MUST** be present.

SlideId: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) element that specifies the identifier of the **presentation slide**. This element **MUST** be present.

SlideIndex: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) element that specifies the zero-based ordered index of the presentation slide. This element **MUST** be present if **SlideId** is equal to 0. This element **MUST** be ignored if **SlideId** is not equal to 0.

2.2.4.8 ServiceError

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface>

A complex type that specifies error information returned from the protocol server to a protocol client.

```

<xs:complexType name="ServiceError" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="RecommendedActions" type="tns1:ClientActions"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Type" type="tns1:ServiceErrorType"/>
  </xs:sequence>
</xs:complexType>

```

Message: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the error message description. The string length **MUST** be greater than zero if the **Type** element has a value of **ApplicationError**. This element **MUST** be present.

RecommendedActions: This element is reserved and **MUST** be ignored.

Title: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the error title. The string length **MUST** be greater than zero if the **Type** element has a value of **ApplicationError**. This element **MUST** be present.

Type: A **ServiceErrorType** (section [2.2.5.11](#)) element that specifies the error type. This element **MUST** be present.

2.2.4.9 ServiceResult

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface

A complex type that specifies the result of a protocol method. The protocol server returns this type to the protocol client containing either a successful **Result** element or an **Error** element.

```
<xs:complexType name="ServiceResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Error" nillable="true" type="tnsl:ServiceError"/>
    <xs:element minOccurs="0" name="Result" nillable="true" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
```

Error: An optional **ServiceError** (section 2.2.4.8) element that specifies an error result of a protocol message response. This element **MUST** be NULL if the **Result** element is not NULL.

Result: An optional **xs:anyType** ([XMLSCHEMA1] section 3.4.7) element that specifies a successful result of a protocol message response. This element **MUST** be NULL if the **Error** element is not NULL.

2.2.4.10 WordStateData

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A complex type that specifies data about the state of a broadcast session of a document.

```
<xs:complexType name="WordStateData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" name="Offset" type="xs:double"/>
    <xs:element minOccurs="0" name="Page" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

Offset: An **xs:double** ([XMLSCHEMA2] section 3.2.5) element that specifies a percentage offset from the top of the page defined by the **Page** field. This element **MUST** be present.

Page: An **xs:int** ([XMLSCHEMA2] section 3.3.17) element that specifies the one-based ordered index of the current page within the set of all pages of the document. This element **MUST** be present.

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
AppType	A simple type that specifies an enumeration of all the possible broadcast session types.

Simple type	Description
BroadcastAppCapability	This simple type is reserved and MUST be ignored.
BroadcastDataKey	A simple type that specifies an enumeration of a set of keys to data stored in the protocol server for a given session.
BroadcastState	A simple type that specifies an enumeration of all the possible broadcast session states.
char	A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.
ClientActions	A simple type that is reserved and MUST be ignored.
duration	A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.
guid	A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.
PPTMediaPlaybackState	A simple type that specifies an enumeration of the different states a multimedia object can be in.
PPTSlideShowState	A simple type that specifies an enumeration of all the possible slide show states.
ServiceErrorType	A simple type that specifies an enumeration of a set of protocol errors returned by the protocol server to the protocol client.

2.2.5.1 AppType

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data>

A simple type that specifies an enumeration of all the possible broadcast session types.

```
<xs:simpleType name="AppType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PPT"/>
    <xs:enumeration value="Word"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **AppType** simple type.

Value	Meaning
PPT	The broadcast session is of a presentation.
Word	The broadcast session is of a document.

2.2.5.2 BroadcastAppCapability

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

This simple type is reserved and MUST be ignored.

```
<xs:simpleType name="BroadcastAppCapability" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SessionTimeout"/>
    <xs:enumeration value="SessionIdleTimeOut"/>
    <xs:enumeration value="SupportVideo"/>
    <xs:enumeration value="SupportAudio"/>
    <xs:enumeration value="SupportNotes"/>
    <xs:enumeration value="MediaExtensions"/>
    <xs:enumeration value="MaxMediaSize"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **BroadcastAppCapability** simple type.

Value	Meaning
SessionTimeout	Reserved and MUST be ignored.
SessionIdleTimeOut	Reserved and MUST be ignored.
SupportVideo	Reserved and MUST be ignored.
SupportAudio	Reserved and MUST be ignored.
SupportNotes	Reserved and MUST be ignored.
MediaExtensions	Reserved and MUST be ignored.
MaxMediaSize	Reserved and MUST be ignored.

2.2.5.3 BroadcastDataKey

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A simple type that specifies an enumeration of a set of keys to data stored in the protocol server for a given session.

```
<xs:simpleType name="BroadcastDataKey" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AppType"/>
    <xs:enumeration value="SequenceNumber"/>
    <xs:enumeration value="FileVersion"/>
    <xs:enumeration value="OriginalFileName"/>
    <xs:enumeration value="BroadcastState"/>
    <xs:enumeration value="AppSpecificStateData"/>
    <xs:enumeration value="NotesUrl"/>
    <xs:enumeration value="DataVersion"/>
  </xs:restriction>
</xs:simpleType>
```

</xs:simpleType>

The following table specifies the allowable values for the **BroadcastDataKey** simple type.

Value	Meaning
AppType	This key is used to specify an application type in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element (section 2.2.4.1).
SequenceNumber	This key is used to specify a monotonically increasing sequence number in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
FileVersion	This key is used to specify a monotonically increasing file version number in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
OriginalFileName	This key is used to specify the file name of the document being shared in the broadcast session in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
BroadcastState	This key is used to specify a BroadcastState (section 2.2.5.4) for the broadcast session in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
AppSpecificStateData	This key is used to specify application specific data for the broadcast session in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
NotesUrl	This key is used to specify the URL of the notes page linked to the document being broadcast in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.
DataVersion	This key is used to specify a data version field in the Key field of a KeyValueOfstringstring element in an ArrayOfKeyValueOfstringstring element.

2.2.5.4 BroadcastState

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A simple type that specifies an enumeration of all the possible broadcast session states.

```
<xs:simpleType name="BroadcastState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BroadcastNotStartedYet"/>
    <xs:enumeration value="BroadcastStarted"/>
    <xs:enumeration value="BroadcastEnded"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **BroadcastState** simple type.

Value	Meaning
BroadcastNotStartedYet	The broadcast session has not started yet.
BroadcastStarted	The broadcast session has started but not ended yet.
BroadcastEnded	The broadcast session has ended.

2.2.5.5 char

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.

```
<xs:simpleType name="char" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
```

2.2.5.6 ClientActions

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface

A simple type that is reserved and MUST be ignored.

```
<xs:simpleType name="ClientActions" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Dismiss"/>
        <xs:enumeration value="Close"/>
        <xs:enumeration value="OpenInClient"/>
        <xs:enumeration value="Refresh"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
```

The following table specifies the allowable values for the **ClientActions** simple type.

Value	Meaning
None	Reserved and MUST be ignored.
Dismiss	Reserved and MUST be ignored.
Close	Reserved and MUST be ignored.
OpenInClient	Reserved and MUST be ignored.
Refresh	Reserved and MUST be ignored.

2.2.5.7 duration

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.

```
<xs:simpleType name="duration" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?" />
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
  </xs:restriction>
</xs:simpleType>
```

2.2.5.8 guid

Namespace: http://schemas.microsoft.com/2003/10/Serialization/

A simple type that is reserved. The protocol client and protocol server MUST NOT use this as the type of any element. The protocol client and protocol server MUST ignore it if receiving an element of this type.

```
<xs:simpleType name="guid" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:pattern value="\da-fA-F}{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
  </xs:restriction>
</xs:simpleType>
```

2.2.5.9 PPTMediaPlaybackState

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data

A simple type that specifies an enumeration of the different states a multimedia object can be in.

```
<xs:simpleType name="PPTMediaPlaybackState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Paused" />
    <xs:enumeration value="Playing" />
    <xs:enumeration value="Stopped" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **PPTMediaPlaybackState** simple type.

Value	Meaning
Paused	The multimedia object is in a paused state.
Playing	The multimedia object is in a playing state.

Value	Meaning
Stopped	The multimedia object is in a stopped state.

2.2.5.10 PPTSlideShowState

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data>

A simple type that specifies an enumeration of all the possible slide show states.

```
<xs:simpleType name="PPTSlideShowState" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BlackScreen"/>
    <xs:enumeration value="WhiteScreen"/>
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="SlideShowEnded"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **PPTSlideShowState** simple type.

Value	Meaning
BlackScreen	The slide show is displaying a black screen.
WhiteScreen	The slide show is displaying a white screen.
Normal	The slide show is displaying presentation slides.
SlideShowEnded	The slide show has ended.

2.2.5.11 ServiceErrorType

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface>

A simple type that specifies an enumeration of a set of protocol errors returned by the protocol server to the protocol client.

```
<xs:simpleType name="ServiceErrorType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UnknownError"/>
    <xs:enumeration value="ApplicationError"/>
    <xs:enumeration value="Timeout"/>
    <xs:enumeration value="ServiceBusy"/>
    <xs:enumeration value="SessionFull"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **ServiceErrorType** simple type.

Value	Meaning
UnknownError	The protocol server encountered an unknown error.
ApplicationError	The protocol server encountered an application error.
Timeout	The protocol server timed out.
ServiceBusy	The protocol server is busy.
SessionFull	The protocol cannot allow more attendees to join a broadcast session.

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

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 that are made by the higher-layer protocol or application are passed directly to the transport, and the results that are returned by the transport are passed directly to the higher-layer protocol or application.

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

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

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

3.1 Server Details

The following high-level sequence diagram illustrates the operation of the participant client protocol.

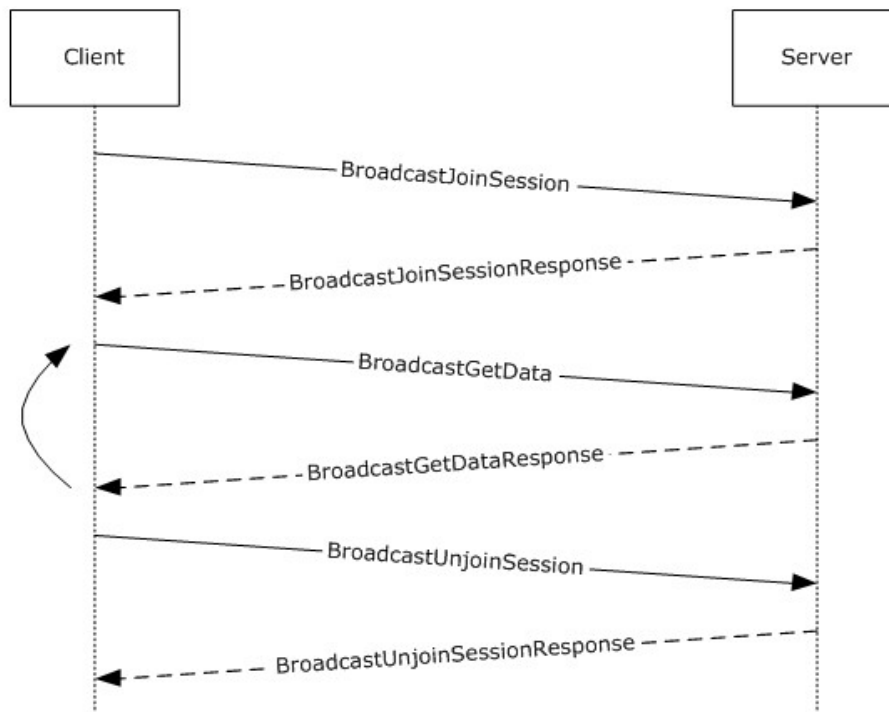


Figure 2: Office Broadcast Participant Service high-level sequence diagram for participant clients

The operation begins when a protocol client acting as broadcast session participant sends a **BroadcastJoinSession** (section [3.1.4.2](#)) message. The server responds with a **BroadcastJoinSessionResponse** message indicating the results of the operation. If the join is successful, the protocol client sends one or more **BroadcastGetData** (section [3.1.4.1](#)) messages, and the protocol server responds to each with a **BroadcastGetDataResponse** message containing the current state of the broadcast session. When the participant client wishes to leave the broadcast, the protocol client sends a **BroadcastUnjoinSession** (section [3.1.4.3](#)) message and the protocol server responds with a **BroadcastUnjoinSessionResponse** message to acknowledge the request.

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.

Note that the abstract interface notation "(Public)" indicates that the abstract data model element can be directly accessed from outside this protocol.

SessionId (Public): An entity that represents a unique identifier for a broadcast session.

BroadcastState (Public): An entity that represents the state of the broadcast session.

AppSpecificStateData (Public): An entity that represents state specific to the application for which this protocol is being used.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
BroadcastGetData	The BroadcastGetData operation is used by the protocol client to retrieve information about the current state of the broadcast session on the protocol server.
BroadcastJoinSession	The BroadcastJoinSession operation is used to join a protocol client to a broadcast session.
BroadcastPing	The BroadcastPing operation is used by a protocol client to check if the protocol server is available.
BroadcastUnjoinSession	The BroadcastUnjoinSession operation is used to disjoin a protocol client from a broadcast session.

3.1.4.1 BroadcastGetData

The **BroadcastGetData** operation is used by the protocol client to retrieve information about the current state of the broadcast session on the protocol server.

```
<wsdl:operation name="BroadcastGetData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastGetData"
    message="tns4:IParticipantService_BroadcastGetData_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastGetDataResponse"
    message="tns4:IParticipantService_BroadcastGetData_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastGetDataSoapIn** request message, and the protocol server MUST respond with a **BroadcastGetDataSoapOut** 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
IParticipantService_BroadcastGetData_InputMessage	The request WSDL message for the BroadcastGetData WSDL operation.
IParticipantService_BroadcastGetData_OutputMessage	The response WSDL message for the BroadcastGetData WSDL operation.

3.1.4.1.1.1 IParticipantService_BroadcastGetData_InputMessage

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastGetData
```

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

3.1.4.1.1.2 IParticipantService_BroadcastGetData_OutputMessage

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

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

3.1.4.1.2 Elements

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

Element	Description
BroadcastGetData	The input data for the BroadcastGetData WSDL operation.
BroadcastGetDataResponse	The result data for the BroadcastGetData WSDL operation.

3.1.4.1.2.1 BroadcastGetData

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

```
<xs:element name="BroadcastGetData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
        xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data"
        minOccurs="0" name="user" nillable="true" type="tns:BroadcastUser"/>
      <xs:element minOccurs="0" name="sequenceNumber" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

user: A **BroadcastUser** (section [3.1.4.1.3.1](#)) element that is obtained by making a **BroadcastJoinSession** (section [3.1.4.2.2.1](#)) web method call. This element **MUST** be present.

sequenceNumber: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) element that specifies the sequence number of the data returned by the previous **BroadcastGetData** (section [3.1.4.1.2.1](#)) operation, if any. A value of 0 **MUST** be passed if this is the first **BroadcastGetData** operation performed by the protocol client. A value of 0 **MAY** be passed at any time. This element **MUST** be present.

3.1.4.1.2.2 BroadcastGetDataResponse

The **BroadcastGetDataResponse** element specifies the result data for the **BroadcastGetData** WSDL operation.

```
<xs:element name="BroadcastGetDataResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
        xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"
        minOccurs="0" name="BroadcastGetDataResult" nillable="true"
        type="tns1:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastGetDataResult: A **ServiceResult** (section [2.2.4.9](#)) element that specifies the result of the operation. This element **MUST** be present.

If the **Error** child element of the **ServiceResult** element is not NULL, the **Result** child element **MUST** be NULL.

If the **Error** child element of the **ServiceResult** element is NULL, the **Result** child element **MUST** follow the following specified behavior:

1. If the **sequenceNumber** parameter to the **BroadcastGetData** (section [3.1.4.1.2.1](#)) operation is nonzero and is greater than or equal to the sequence number most recently received by the protocol server from an Office Broadcast Presentation Service [\[MS-OBPRS\]](#) protocol client, the **Result** child element MUST NOT be present. This indicates to the protocol client that the state data of the broadcast session has not changed.
2. In all other cases, the **Result** child element MUST be present and MUST be an **ArrayOfKeyValueOfStringString** (section [2.2.4.1](#)) element. The **ArrayOfKeyValueOfStringString** contains **KeyValueOfStringString** child elements where the **Key** child element MUST be a **BroadcastDataKey** (section [2.2.5.3](#)) enumeration value of type **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1). The **Value** child of a **KeyValueOfStringString** element MUST conform to the following table, and all the values are returned as **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) format.

Key	Values
AppType	MUST be an AppType (section 2.2.5.2) enumeration value of type xs:string ([XMLSCHEMA2] section 3.2.1) element.
SequenceNumber	MUST be an xs:string ([XMLSCHEMA2] section 3.2.1) element reformatted from an xs:int ([XMLSCHEMA2] section 3.3.17) element.
FileVersion	MUST be an xs:string ([XMLSCHEMA2] section 3.2.1) element reformatted from an xs:int ([XMLSCHEMA2] section 3.3.17) element.
OriginalFileName	MUST be an xs:string ([XMLSCHEMA2] section 3.2.1) element.
BroadcastState	MUST be a BroadcastState (section 2.2.5.10) enumeration value of type xs:string ([XMLSCHEMA2] section 3.2.1) element.
AppSpecificStateData	MUST be a JSON [RFC4627] serialization of a PPTStateData (section 2.2.4.7) element if the AppType is PPT or a JSON [RFC4627] serialization of a WordStateData (section 2.2.4.10) element if the AppType is Word .
NotesUrl	MUST be an xs:string ([XMLSCHEMA2] section 3.2.1) element.
DataVersion	MUST be an xs:string ([XMLSCHEMA2] section 3.2.1) element reformatted from an xs:int ([XMLSCHEMA2] section 3.3.17) element with a value of "2".

3.1.4.1.3 Complex Types

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

Complex type	Description
BroadcastUser	A complex type that specifies a user of a broadcast session.

3.1.4.1.3.1 BroadcastUser

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data>

A complex type that specifies a user of a broadcast session.

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

```

<xs:sequence>
  <xs:element minOccurs="0" name="SessionId" nillable="true" type="xs:string"/>
  <xs:element minOccurs="0" name="UserToken" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>

```

SessionId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the identifier of the broadcast session on the protocol server. This element **MUST** be present.

UserToken: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the identifier of a user of the broadcast session on the protocol server. This element **MUST** be present.

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 BroadcastJoinSession

The **BroadcastJoinSession** operation is used to join a protocol client to a broadcast session.

```

<wsdl:operation name="BroadcastJoinSession" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastJoinSession"
    message="tns4:IParticipantService_BroadcastJoinSession_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastJoinSessionResponse"
    message="tns4:IParticipantService_BroadcastJoinSession_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends a **BroadcastJoinSessionSoapIn** request message, and the protocol server **MUST** respond with a **BroadcastJoinSessionSoapOut** 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
IParticipantService_BroadcastJoinSession_InputMessage	The request WSDL message for the BroadcastJoinSession WSDL operation.
IParticipantService_BroadcastJoinSession_OutputMessage	The response WSDL message for the BroadcastJoinSession WSDL operation.

3.1.4.2.1.1 IParticipantService_BroadcastJoinSession_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/
IParticipantService/BroadcastJoinSession
```

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

3.1.4.2.1.2 IParticipantService_BroadcastJoinSession_OutputMessage

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

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

3.1.4.2.2 Elements

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

Element	Description
BroadcastJoinSession	The input data for the BroadcastJoinSession WSDL operation.
BroadcastJoinSessionResponse	The result data for the BroadcastJoinSession WSDL operation.

3.1.4.2.2.1 BroadcastJoinSession

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

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

sessionId: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) element that specifies an identifier for the broadcast session. This identifier **MUST** conform to the pattern **WOPISrc=<source>&access_token=<token>**, where **<source>** is the **WOPISrc** parameter as

defined in [\[MS-WOPI\]](#) section 3.1.5.1.1.2.3.3, and **<token>** is the <token> parameter as defined in [\[MS-WOPI\]](#) section 2.2.2.

3.1.4.2.2.2 BroadcastJoinSessionResponse

The **BroadcastJoinSessionResponse** element specifies the result data for the **BroadcastJoinSession** WSDL operation.

```
<xs:element name="BroadcastJoinSessionResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
        xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastJoinSessionResult" nillable="true"
        type="tns1:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastJoinSessionResult: A **ServiceResult** (section [2.2.4.9](#)) element that specifies the result of the operation. This element **MUST** be present. The **Result** child element **MUST** be a **BroadcastUser** (section [3.1.4.1.3.1](#)) if the **Error** child element is not present.

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 BroadcastUnjoinSession

The **BroadcastUnjoinSession** operation is used to disjoin a protocol client from a broadcast session.

```
<wsdl:operation name="BroadcastUnjoinSession" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastUnjoinSession"
    message="tns4:IParticipantService_BroadcastUnjoinSession_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
```

```

    <wsdl:output
      wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastUnjoinSessionResponse"
      message="tns4:IParticipantService_BroadcastUnjoinSession_OutputMessage"
      xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:operation>

```

The protocol client sends a **BroadcastUnjoinSessionSoapIn** request message, and the protocol server MUST respond with a **BroadcastUnjoinSessionSoapOut** 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
IParticipantService_BroadcastUnjoinSession_InputMessage	The request WSDL message for the BroadcastUnjoinSession WSDL operation.
IParticipantService_BroadcastUnjoinSession_OutputMessage	The response WSDL message for the BroadcastUnjoinSession WSDL operation.

3.1.4.3.1.1 IParticipantService_BroadcastUnjoinSession_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastUnjoinSession

```

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

3.1.4.3.1.2 IParticipantService_BroadcastUnjoinSession_OutputMessage

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

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

3.1.4.3.2 Elements

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

Element	Description
BroadcastUnjoinSession	The input data for the BroadcastUnjoinSession WSDL operation.
BroadcastUnjoinSessionResponse	The result data for the BroadcastUnjoinSession WSDL operation.

3.1.4.3.2.1 BroadcastUnjoinSession

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

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

sessionId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the broadcast session to disjoin from. This element **MUST** be present.

3.1.4.3.2.2 BroadcastUnjoinSessionResponse

The **BroadcastUnjoinSessionResponse** element specifies the result data for the **BroadcastUnjoinSession** WSDL operation.

```
<xs:element name="BroadcastUnjoinSessionResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element
        xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"
        minOccurs="0" name="BroadcastUnjoinSessionResult" nillable="true"
        type="tns1:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastUnjoinSessionResult: A **ServiceResult** (section [2.2.4.9](#)) element that specifies the result of the operation. This element **MUST** be present. If the **Result** child element is present, it **MUST** be ignored by the protocol client.

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 BroadcastPing

The **BroadcastPing** operation is used by a protocol client to check if the protocol server is available.

```
<wsdl:operation name="BroadcastPing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastPing"
    message="tns4:IParticipantService_BroadcastPing_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastPingResponse"
    message="tns4:IParticipantService_BroadcastPing_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastPingSoapIn** request message, as specified in section [3.1.4.4.1.1](#), and the protocol server MUST respond with a **BroadcastPingSoapOut** response message, as specified in section [3.1.4.4.1.2](#).

3.1.4.4.1 Messages

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

Message	Description
IParticipantService_BroadcastPing_InputMessage	The request WSDL message for the BroadcastPing WSDL operation.
IParticipantService_BroadcastPing_OutputMessage	The response WSDL message for the BroadcastPing WSDL operation.

3.1.4.4.1.1 IParticipantService_BroadcastPing_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastPing
```

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

3.1.4.4.1.2 IParticipantService_BroadcastPing_OutputMessage

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

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

3.1.4.4.2 Elements

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

Element	Description
BroadcastPing	The input data for the BroadcastPing WSDL operation.
BroadcastPingResponse	The result data for the BroadcastPing WSDL operation.

3.1.4.4.2.1 BroadcastPing

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

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

3.1.4.4.2.2 BroadcastPingResponse

The **BroadcastPingResponse** element specifies the result data for the **BroadcastPing** WSDL operation.

```
<xs:element name="BroadcastPingResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="BroadcastPingResult" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastPingResult: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) element. This element MUST be present and MUST be set to true.

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples contain sample interactions between protocol clients and protocol servers.

4.1 Attendee Client Example

The attendee protocol client begins by sending a request to the protocol server to join the broadcast session. The following **BroadcastJoinSessionSoapIn** (section [3.1.4.2](#)) message is sent to the protocol server:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <BroadcastJoinSession
      xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantS
      ervice/">
      <sessionId>WOPISrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access
      _token=VAR</sessionId>
    </BroadcastJoinSession>
  </s:Body>
</s:Envelope>
```

The protocol server responds with a message to acknowledge the request and to provide host information and an identifier for the broadcast session to be used by the attendee protocol client for future requests. The following **BroadcastJoinSessionSoapOut** (section [3.1.4.2](#)) message is sent to the attendee protocol client:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <BroadcastJoinSessionResponse
      xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantS
      ervice/">
      <BroadcastJoinSessionResult
        xmlns:a="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Inter
        face" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Error i:nil="true"/>
        <a:Result i:type="b:BroadcastUser"
          xmlns:b="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.
          Data">
          <b:SessionId>WOPISrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&acce
          ss_token=VAR</b:SessionId>
          <b:UserToken>b5ea6891-bb86-4544-a74c-847ce4c2edee</b:UserToken>
        </a:Result>
      </BroadcastJoinSessionResult>
    </BroadcastJoinSessionResponse>
  </s:Body>
</s:Envelope>
```

Next, the attendee protocol client requests information about the current state of the broadcast session from the protocol server. The following **BroadcastGetDataSoapIn** (section [3.1.4.1](#)) message is sent to the protocol server:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <BroadcastGetData
      xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantS
      ervice/">

```

```

<user
  xmlns:a="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.
  Data" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <a:SessionId>WOPISrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&acces
  ss_token=VAR</a:SessionId>
  <a:UserToken>b5ea6891-bb86-4544-a74c-847ce4c2edee</a:UserToken>
</user>
<sequenceNumber>0</sequenceNumber>
</BroadcastGetData>
</s:Body>
</s:Envelope>

```

The protocol server responds with information about the current state of the broadcast session. The following **BroadcastGetDataSoapOut** (section [3.1.4.1](#)) message is sent to the attendee protocol client:

```

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body><BroadcastGetDataResponse
  xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantS
  ervice/">
  <BroadcastGetDataResult
  xmlns:a="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Inter
  face" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <a:Error i:nil="true"/>
  <a:Result i:type="b:ArrayOfKeyValueOfstringstring"
  xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
  <b:KeyValueOfstringstring>
  <b:Key>AppType</b:Key>
  <b:Value>PPT</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>BroadcastState</b:Key>
  <b:Value>BroadcastStarted</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>SequenceNumber</b:Key>
  <b:Value>5</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>FileVersion</b:Key>
  <b:Value>1</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>OriginalFileName</b:Key>
  <b:Value>file.pptx</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>NotesUrl</b:Key>
  <b:Value>http://url</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>DataVersion</b:Key>
  <b:Value>2</b:Value>
  </b:KeyValueOfstringstring>
  <b:KeyValueOfstringstring>
  <b:Key>AppSpecificStateData</b:Key>
  <b:Value>{"SlideId":258,"SlideIndex":0,"AnimationStepDataList":[{"TimelineId":"timeline_1","S
  tep":1}, {"TimelineId":"timeline_2","Step":2}], "MediaStateDataList":[{"MediaId":"306s4f1","Sta

```

```

te":1,"AtPosition":0.015},{ "MediaId":"55s21","State":0,"AtPosition":34.53}]], "PPTSlideShowState":2}</b:Value>
</b:KeyValueOfstringstring>
</a:Result>
</BroadcastGetDataResult>
</BroadcastGetDataResponse>
</s:Body>
</s:Envelope>

```

The attendee protocol client continues to send **BroadcastGetDataSoapIn** (section [3.1.4.1](#)) messages to request the latest state of the broadcast session.

Finally, the attendee protocol client disjoins from the broadcast session by sending the **BroadcastUnjoinSessionSoapIn** (section [3.1.4.3](#)) message to the protocol server:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body>
<BroadcastUnjoinSession
xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantsService/">
<sessionId>WOPISrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access_token=VAR</sessionId>
</BroadcastUnjoinSession>
</s:Body>
</s:Envelope>

```

The protocol server responds by sending the **BroadcastUnjoinSessionSoapOut** (section [3.1.4.3](#)) message to the attendee protocol client:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Body>
<BroadcastUnjoinSessionResponse
xmlns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantsService/">
<BroadcastUnjoinSessionResult
xmlns:a="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<a:Error i:nil="true"/>
<a:Result i:nil="true"/>
</BroadcastUnjoinSessionResult>
</BroadcastUnjoinSessionResponse>
</s:Body>
</s:Envelope>

```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```
<?xml version="1.0"?>
<wsdl:definitions
  xmlns:tns4="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
  pantService/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPa
  rticipantService/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema
      xmlns:tns5="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
      pantService/Imports"
      targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPa
      rticipantService/Imports">
      <xs:import
        namespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticip
        antService/">
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/">
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interfac
        e.Data"/>
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Int
        erface"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
      </xs:schema>
    </wsdl:types>
    <wsdl:message name="IParticipantService_BroadcastJoinSession_InputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastJoinSession"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastJoinSession_OutputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastJoinSessionResponse"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastUnjoinSession_InputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastUnjoinSession"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastUnjoinSession_OutputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastUnjoinSessionResponse"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastGetData_InputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastGetData"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastGetData_OutputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastGetDataResponse"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastPing_InputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastPing"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastPing_OutputMessage">
      <wsdl:part name="parameters" element="tns4:BroadcastPingResponse"/>
    </wsdl:message>
    <wsdl:portType name="IParticipantService">
      <wsdl:operation name="BroadcastJoinSession">
        <wsdl:input
          wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
```

```

    ipantService/IParticipantService/BroadcastJoinSession"
    message="tns4:IParticipantService_BroadcastJoinSession_InputMessage"/>
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastJoinSessionResponse"
    message="tns4:IParticipantService_BroadcastJoinSession_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastUnjoinSession">
    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastUnjoinSession"
    message="tns4:IParticipantService_BroadcastUnjoinSession_InputMessage"/>
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastUnjoinSessionResponse"
    message="tns4:IParticipantService_BroadcastUnjoinSession_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastGetData">
    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastGetData"
    message="tns4:IParticipantService_BroadcastGetData_InputMessage"/>
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastGetDataResponse"
    message="tns4:IParticipantService_BroadcastGetData_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastPing">
    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastPing"
    message="tns4:IParticipantService_BroadcastPing_InputMessage"/>
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartic
ipantService/IParticipantService/BroadcastPingResponse"
    message="tns4:IParticipantService_BroadcastPing_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding_IParticipantService" type="tns4:IParticipantService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="BroadcastJoinSession">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
pantService/IParticipantService/BroadcastJoinSession" style="document"/>
    <wsdl:input>
    <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
    <soap:body use="literal"/>
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="BroadcastUnjoinSession">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
pantService/IParticipantService/BroadcastUnjoinSession" style="document"/>
    <wsdl:input>
    <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
    <soap:body use="literal"/>
    </wsdl:output>
    </wsdl:operation>

```

```

    <wsdl:operation name="BroadcastGetData">
      <soap:operation
        soapAction="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
        pantService/IParticipantService/BroadcastGetData" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="BroadcastPing">
      <soap:operation
        soapAction="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
        pantService/IParticipantService/BroadcastPing" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>

```

7 Appendix B: Full XML Schema

For ease of implementation, the following sections provide the full XML schema for this protocol.

Schema name	Prefix	Section
http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data	tns	7.1
http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface	tns1	7.2
http://schemas.microsoft.com/2003/10/Serialization/Arrays	tns2	7.3
http://schemas.microsoft.com/2003/10/Serialization/	tns3	7.4
http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/	tns4	7.5

7.1

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data> Schema

```
<?xml version="1.0"?>
<xs:schema
  xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data" elementFormDefault="qualified"
  targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BroadcastUser">
    <xs:sequence>
      <xs:element minOccurs="0" name="SessionId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="UserToken" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="BroadcastUser" nillable="true" type="tns:BroadcastUser"/>
  <xs:complexType name="PPTStateData">
    <xs:sequence>
      <xs:element minOccurs="0" name="AnimationStepDataList" nillable="true"
type="tns:ArrayOfPPTBroadcastAnimationStepData"/>
      <xs:element minOccurs="0" name="MediaStateDataList" nillable="true"
type="tns:ArrayOfPPTBroadcastMediaStateData"/>
      <xs:element minOccurs="0" name="PPTSlideShowState" type="tns:PPTSlideShowState"/>
      <xs:element minOccurs="0" name="SlideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="SlideIndex" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="PPTStateData" nillable="true" type="tns:PPTStateData"/>
  <xs:complexType name="ArrayOfPPTBroadcastAnimationStepData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="PPTBroadcastAnimationStepData"
nillable="true" type="tns:PPTBroadcastAnimationStepData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfPPTBroadcastAnimationStepData" nillable="true"
type="tns:ArrayOfPPTBroadcastAnimationStepData"/>
</xs:schema>
```

```

<xs:complexType name="PPTBroadcastAnimationStepData">
  <xs:sequence>
    <xs:element minOccurs="0" name="Step" type="xs:int"/>
    <xs:element minOccurs="0" name="TimelineId" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PPTBroadcastAnimationStepData" nillable="true"
type="tns:PPTBroadcastAnimationStepData"/>
<xs:complexType name="ArrayOfPPTBroadcastMediaStateData">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="PPTBroadcastMediaStateData"
nillable="true" type="tns:PPTBroadcastMediaStateData"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfPPTBroadcastMediaStateData" nillable="true"
type="tns:ArrayOfPPTBroadcastMediaStateData"/>
<xs:complexType name="PPTBroadcastMediaStateData">
  <xs:sequence>
    <xs:element minOccurs="0" name="AtPosition" type="xs:double"/>
    <xs:element minOccurs="0" name="MediaId" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="State" type="tns:PPTMediaPlaybackState"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PPTBroadcastMediaStateData" nillable="true"
type="tns:PPTBroadcastMediaStateData"/>
<xs:simpleType name="PPTMediaPlaybackState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Paused"/>
    <xs:enumeration value="Playing"/>
    <xs:enumeration value="Stopped"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="PPTMediaPlaybackState" nillable="true" type="tns:PPTMediaPlaybackState"/>
<xs:simpleType name="PPTSlideShowState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BlackScreen"/>
    <xs:enumeration value="WhiteScreen"/>
    <xs:enumeration value="Normal"/>
    <xs:enumeration value="SlideShowEnded"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="PPTSlideShowState" nillable="true" type="tns:PPTSlideShowState"/>
<xs:complexType name="WordStateData">
  <xs:sequence>
    <xs:element minOccurs="0" name="Offset" type="xs:double"/>
    <xs:element minOccurs="0" name="Page" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="WordStateData" nillable="true" type="tns:WordStateData"/>
<xs:complexType name="EnumTypes">
  <xs:sequence>
    <xs:element minOccurs="0" name="AppType" type="tns:AppType"/>
    <xs:element minOccurs="0" name="BroadcastAppCapability"
type="tns:BroadcastAppCapability"/>
    <xs:element minOccurs="0" name="BroadcastDataKey" type="tns:BroadcastDataKey"/>
    <xs:element minOccurs="0" name="BroadcastState" type="tns:BroadcastState"/>
    <xs:element minOccurs="0" name="PPTMediaPlaybackState"
type="tns:PPTMediaPlaybackState"/>
    <xs:element minOccurs="0" name="PPTSlideShowState" type="tns:PPTSlideShowState"/>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
</xs:complexType>
<xs:element name="EnumTypes" nillable="true" type="tns:EnumTypes"/>
<xs:simpleType name="AppType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="PPT"/>
    <xs:enumeration value="Word"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="AppType" nillable="true" type="tns:AppType"/>
<xs:simpleType name="BroadcastAppCapability">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SessionTimeout"/>
    <xs:enumeration value="SessionIdleTimeOut"/>
    <xs:enumeration value="SupportVideo"/>
    <xs:enumeration value="SupportAudio"/>
    <xs:enumeration value="SupportNotes"/>
    <xs:enumeration value="MediaExtensions"/>
    <xs:enumeration value="MaxMediaSize"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="BroadcastAppCapability" nillable="true"
type="tns:BroadcastAppCapability"/>
<xs:simpleType name="BroadcastDataKey">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AppType"/>
    <xs:enumeration value="SequenceNumber"/>
    <xs:enumeration value="FileVersion"/>
    <xs:enumeration value="OriginalFileName"/>
    <xs:enumeration value="BroadcastState"/>
    <xs:enumeration value="AppSpecificStateData"/>
    <xs:enumeration value="NotesUrl"/>
    <xs:enumeration value="DataVersion"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="BroadcastDataKey" nillable="true" type="tns:BroadcastDataKey"/>
<xs:simpleType name="BroadcastState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BroadcastNotStartedYet"/>
    <xs:enumeration value="BroadcastStarted"/>
    <xs:enumeration value="BroadcastEnded"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="BroadcastState" nillable="true" type="tns:BroadcastState"/>
</xs:schema>

```

7.2

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface Schema

```

<?xml version="1.0"?>
<xs:schema
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/">

```

```

<xs:complexType name="ServiceResult">
  <xs:sequence>
    <xs:element minOccurs="0" name="Error" nillable="true" type="tnsl:ServiceError"/>
    <xs:element minOccurs="0" name="Result" nillable="true" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ServiceResult" nillable="true" type="tnsl:ServiceResult"/>
<xs:complexType name="ServiceError">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="RecommendedActions" type="tnsl:ClientActions"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Type" type="tnsl:ServiceErrorType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ServiceError" nillable="true" type="tnsl:ServiceError"/>
<xs:simpleType name="ClientActions">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Dismiss"/>
        <xs:enumeration value="Close"/>
        <xs:enumeration value="OpenInClient"/>
        <xs:enumeration value="Refresh"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
<xs:element name="ClientActions" nillable="true" type="tnsl:ClientActions"/>
<xs:simpleType name="ServiceErrorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UnknownError"/>
    <xs:enumeration value="ApplicationError"/>
    <xs:enumeration value="Timeout"/>
    <xs:enumeration value="ServiceBusy"/>
    <xs:enumeration value="SessionFull"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="ServiceErrorType" nillable="true" type="tnsl:ServiceErrorType"/>
</xs:schema>

```

7.3 <http://schemas.microsoft.com/2003/10/Serialization/Arrays> Schema

```

<?xml version="1.0"?>
<xs:schema xmlns:tns2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="ArrayOfKeyValueOfstringstring">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="KeyValueOfstringstring">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Key" nillable="true" type="xs:string"/>
            <xs:element name="Value" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfKeyValueOfstringstring" nillable="true"
type="tns2:ArrayOfKeyValueOfstringstring"/>
</xs:schema>

```

7.4 http://schemas.microsoft.com/2003/10/Serialization/ Schema

```

<?xml version="1.0"?>
<xs:schema xmlns:tns3="http://schemas.microsoft.com/2003/10/Serialization/"
attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="anyType" nillable="true" type="xs:anyType"/>
    <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
    <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
    <xs:element name="boolean" nillable="true" type="xs:boolean"/>
    <xs:element name="byte" nillable="true" type="xs:byte"/>
    <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
    <xs:element name="decimal" nillable="true" type="xs:decimal"/>
    <xs:element name="double" nillable="true" type="xs:double"/>
    <xs:element name="float" nillable="true" type="xs:float"/>
    <xs:element name="int" nillable="true" type="xs:int"/>
    <xs:element name="long" nillable="true" type="xs:long"/>
    <xs:element name="QName" nillable="true" type="xs:QName"/>
    <xs:element name="short" nillable="true" type="xs:short"/>
    <xs:element name="string" nillable="true" type="xs:string"/>
    <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
    <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
    <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
    <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
    <xs:element name="char" nillable="true" type="tns3:char"/>
    <xs:simpleType name="char">
        <xs:restriction base="xs:int"/>
    </xs:simpleType>
    <xs:element name="duration" nillable="true" type="tns3:duration"/>
    <xs:simpleType name="duration">
        <xs:restriction base="xs:duration">
            <xs:pattern value="\-?P(\d*D)?(T(\d*M)?(\d*(\.\d*)?S)?)?" />
            <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
            <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="guid" nillable="true" type="tns3:guid"/>
    <xs:simpleType name="guid">
        <xs:restriction base="xs:string">
            <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
        </xs:restriction>
    </xs:simpleType>
    <xs:attribute name="FactoryType" type="xs:QName"/>
    <xs:attribute name="Id" type="xs:ID"/>
    <xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>

```


7.5

http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/ Schema

```
<?xml version="1.0"?>
<xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"/>
  <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data"/>
  <xs:element name="BroadcastJoinSession">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="sessionId" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastJoinSessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastJoinSessionResult" nillable="true"
type="tns1:ServiceResult"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastUnjoinSession">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="sessionId" nillable="true" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastUnjoinSessionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastUnjoinSessionResult" nillable="true"
type="tns1:ServiceResult"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastGetData">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data" minOccurs="0" name="user" nillable="true" type="tns:BroadcastUser"/>
        <xs:element minOccurs="0" name="sequenceNumber" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastGetDataResponse">
    <xs:complexType>
```

```

        <xs:sequence>
          <xs:element
xmlns:tnsl="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.In
terface" minOccurs="0" name="BroadcastGetDataResult" nillable="true"
type="tnsl:ServiceResult"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="BroadcastPing">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <xs:element name="BroadcastPingResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" name="BroadcastPingResult" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>

```

8 Appendix C: 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® Lync® Server 2013
- Microsoft® OneNote® 2013

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

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

9 Change Tracking

This section identifies changes that were made to the [MS-OBPAS] protocol document between the July 2012 and October 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.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1 Introduction	Updated all preliminary information in the document.	Y	Content updated.
2.2.4.9 ServiceResult	Changed "MUST be present if the element is not present" to "MUST be NULL if the element is not NULL" in descriptions for the Error and Result elements.	N	Content updated.
3.1.4.1.2.2 BroadcastGetDataResponse	Changed "MUST be present if the element is not present" to "MUST be NULL if the element is not NULL" in descriptions for the Error and Result elements.	N	Content updated.
3.1.4.1.2.2 BroadcastGetDataResponse	Updated the descriptions of the values for a KeyValueOfString element to specify that the values are formatted as type xs:string (including reference).	N	Content updated.
4.1 Attendee Client Example	Update the value of sessionId in first XML message example.	N	Content updated.

10 Index

A

[Abstract data model server](#) 25
[Applicability](#) 8
[AppType simple type](#) 17
[ArrayOfKeyValueOfstringstring complex type](#) 12
[ArrayOfPPTBroadcastAnimationStepData complex type](#) 12
[ArrayOfPPTBroadcastMediaStateData complex type](#) 13
[Attendee Client example](#) 37
[Attribute groups](#) 23
[Attributes](#) 23

B

[BroadcastAppCapability simple type](#) 18
[BroadcastDataKey simple type](#) 18
[BroadcastState simple type](#) 19

C

[Capability negotiation](#) 8
[Change tracking](#) 52
[char simple type](#) 20
[ClientActions simple type](#) 20
[Complex types](#) 11
 [ArrayOfKeyValueOfstringstring](#) 12
 [ArrayOfPPTBroadcastAnimationStepData](#) 12
 [ArrayOfPPTBroadcastMediaStateData](#) 13
 [EnumTypes](#) 13
 [PPTBroadcastAnimationStepData](#) 13
 [PPTBroadcastMediaStateData](#) 14
 [PPTStateData](#) 14
 [ServiceError](#) 15
 [ServiceResult](#) 16
 [WordStateData](#) 16

D

Data model - abstract
 [server](#) 25
[duration simple type](#) 21

E

[EnumTypes complex type](#) 13
Events
 [local - server](#) 36
 [timer - server](#) 36
Examples
 [Attendee Client](#) 37
 [Overview](#) 37

F

[Fields - vendor-extensible](#) 8

[Full WSDL](#) 41
[Full XML Schema](#) 44
 <http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.DataSchema> 44
 <http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.InterfaceSchema> 46
 <http://schemas.microsoft.com/2003/10/Serialization/Schema> 48
 <http://schemas.microsoft.com/2003/10/Serialization/ArraysSchema> 47
 <http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Schema> 49

G

[Glossary](#) 6
[Groups](#) 23
[guid simple type](#) 21

I

[Implementer - security considerations](#) 40
[Index of security parameters](#) 40
[Informative references](#) 7
Initialization
 [server](#) 25
[Introduction](#) 6

L

Local events
 [server](#) 36

M

Message processing
 [server](#) 25
Messages
 [AppType simple type](#) 17
 [ArrayOfKeyValueOfstringstring complex type](#) 12
 [ArrayOfPPTBroadcastAnimationStepData complex type](#) 12
 [ArrayOfPPTBroadcastMediaStateData complex type](#) 13
 [attribute groups](#) 23
 [attributes](#) 23
 [BroadcastAppCapability simple type](#) 18
 [BroadcastDataKey simple type](#) 18
 [BroadcastState simple type](#) 19
 [char simple type](#) 20
 [ClientActions simple type](#) 20
 [complex types](#) 11
 [duration simple type](#) 21
 [elements](#) 11
 [enumerated](#) 11

- [EnumTypes complex type](#) 13
- [groups](#) 23
- [guid simple type](#) 21
- [namespaces](#) 10
- [PPTBroadcastAnimationStepData complex type](#) 13
- [PPTBroadcastMediaStateData complex type](#) 14
- [PPTMediaPlaybackState simple type](#) 21
- [PPTSlideShowState simple type](#) 22
- [PPTStateData complex type](#) 14
- [ServiceError complex type](#) 15
- [ServiceErrorType simple type](#) 22
- [ServiceResult complex type](#) 16
- [simple types](#) 16
- [syntax](#) 10
- [transport](#) 10
- [WordStateData complex type](#) 16

N

- [Namespaces](#) 10
- [Normative references](#) 6

O

- Operations
 - [BroadcastGetData](#) 26
 - [BroadcastJoinSession](#) 29
 - [BroadcastPing](#) 34
 - [BroadcastUnjoinSession](#) 31
- [Overview \(synopsis\)](#) 7

P

- [Parameters - security index](#) 40
- [PPTBroadcastAnimationStepData complex type](#) 13
- [PPTBroadcastMediaStateData complex type](#) 14
- [PPTMediaPlaybackState simple type](#) 21
- [PPTSlideShowState simple type](#) 22
- [PPTStateData complex type](#) 14
- [Preconditions](#) 8
- [Prerequisites](#) 8
- [Product behavior](#) 51

R

- [References](#) 6
 - [informative](#) 7
 - [normative](#) 6
- [Relationship to other protocols](#) 8

S

- Security
 - [implementer considerations](#) 40
 - [parameter index](#) 40
- Sequencing rules
 - [server](#) 25
- Server
 - [abstract data model](#) 25
 - [BroadcastGetData operation](#) 26
 - [BroadcastJoinSession operation](#) 29

- [BroadcastPing operation](#) 34
- [BroadcastUnjoinSession operation](#) 31
- [initialization](#) 25
- [local events](#) 36
- [message processing](#) 25
- [sequencing rules](#) 25
- [timer events](#) 36
- [timers](#) 25
- [ServiceError complex type](#) 15
- [ServiceErrorType simple type](#) 22
- [ServiceResult complex type](#) 16
- [Simple types](#) 16
 - [AppType](#) 17
 - [BroadcastAppCapability](#) 18
 - [BroadcastDataKey](#) 18
 - [BroadcastState](#) 19
 - [char](#) 20
 - [ClientActions](#) 20
 - [duration](#) 21
 - [guid](#) 21
 - [PPTMediaPlaybackState](#) 21
 - [PPTSlideShowState](#) 22
 - [ServiceErrorType](#) 22
- [Standards assignments](#) 9
- Syntax
 - [messages - overview](#) 10

T

- Timer events
 - [server](#) 36
- Timers
 - [server](#) 25
- [Tracking changes](#) 52
- [Transport](#) 10
- Types
 - [complex](#) 11
 - [simple](#) 16

V

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

W

- [WordStateData complex type](#) 16
- [WSDL](#) 41

X

- [XML Schema](#) 44
 - <http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data.Schema> 44
 - <http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface.Schema> 46
 - <http://schemas.microsoft.com/2003/10/Serialization/Schema> 48
 - <http://schemas.microsoft.com/2003/10/Serialization/Arrays.Schema> 47

<http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Schema> 49