

[MS-OBPAS]:

Office Broadcast Participant Service

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation 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 might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
3/16/2015	2.0	Major	Significantly changed the technical content.
6/30/2015	3.0	Major	Significantly changed the technical content.
6/23/2016	4.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
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	12
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	15
2.2.4.10	WordStateData	16
2.2.5	Simple Types	16
2.2.5.1	AppType	17
2.2.5.2	BroadcastAppCapability	17
2.2.5.3	BroadcastDataKey.....	18
2.2.5.4	BroadcastState	19
2.2.5.5	char	19
2.2.5.6	ClientActions	19
2.2.5.7	duration.....	20
2.2.5.8	guid	20
2.2.5.9	PPTMediaPlaybackState	20
2.2.5.10	PPTSlideShowState	21
2.2.5.11	ServiceErrorType	21
2.2.6	Attributes	22
2.2.7	Groups	22
2.2.8	Attribute Groups.....	22
3	Protocol Details.....	23
3.1	Server Details.....	23
3.1.1	Abstract Data Model.....	24
3.1.2	Timers	24
3.1.3	Initialization.....	24
3.1.4	Message Processing Events and Sequencing Rules	24
3.1.4.1	BroadcastGetData	25
3.1.4.1.1	Messages	25
3.1.4.1.1.1	IParticipantService_BroadcastGetData_InputMessage.....	25

3.1.4.1.1.2	IParticipantService_BroadcastGetData_OutputMessage	25
3.1.4.1.2	Elements	25
3.1.4.1.2.1	BroadcastGetData	26
3.1.4.1.2.2	BroadcastGetDataResponse	26
3.1.4.1.3	Complex Types	27
3.1.4.1.3.1	BroadcastUser	27
3.1.4.1.4	Simple Types	28
3.1.4.1.5	Attributes	28
3.1.4.1.6	Groups	28
3.1.4.1.7	Attribute Groups	28
3.1.4.2	BroadcastJoinSession	28
3.1.4.2.1	Messages	28
3.1.4.2.1.1	IParticipantService_BroadcastJoinSession_InputMessage	29
3.1.4.2.1.2	IParticipantService_BroadcastJoinSession_OutputMessage	29
3.1.4.2.2	Elements	29
3.1.4.2.2.1	BroadcastJoinSession	29
3.1.4.2.2.2	BroadcastJoinSessionResponse	29
3.1.4.2.3	Complex Types	30
3.1.4.2.4	Simple Types	30
3.1.4.2.5	Attributes	30
3.1.4.2.6	Groups	30
3.1.4.2.7	Attribute Groups	30
3.1.4.3	BroadcastUnjoinSession	30
3.1.4.3.1	Messages	31
3.1.4.3.1.1	IParticipantService_BroadcastUnjoinSession_InputMessage	31
3.1.4.3.1.2	IParticipantService_BroadcastUnjoinSession_OutputMessage	31
3.1.4.3.2	Elements	31
3.1.4.3.2.1	BroadcastUnjoinSession	31
3.1.4.3.2.2	BroadcastUnjoinSessionResponse	32
3.1.4.3.3	Complex Types	32
3.1.4.3.4	Simple Types	32
3.1.4.3.5	Attributes	32
3.1.4.3.6	Groups	32
3.1.4.3.7	Attribute Groups	32
3.1.4.4	BroadcastPing	32
3.1.4.4.1	Messages	33
3.1.4.4.1.1	IParticipantService_BroadcastPing_InputMessage	33
3.1.4.4.1.2	IParticipantService_BroadcastPing_OutputMessage	33
3.1.4.4.2	Elements	33
3.1.4.4.2.1	BroadcastPing	33
3.1.4.4.2.2	BroadcastPingResponse	34
3.1.4.4.3	Complex Types	34
3.1.4.4.4	Simple Types	34
3.1.4.4.5	Attributes	34
3.1.4.4.6	Groups	34
3.1.4.4.7	Attribute Groups	34
3.1.5	Timer Events	34
3.1.6	Other Local Events	34
4	Protocol Examples	35
4.1	Attendee Client Example	35
5	Security	38
5.1	Security Considerations for Implementers	38
5.2	Index of Security Parameters	38
6	Appendix A: Full WSDL	39
7	Appendix B: Full XML Schema	42

7.1	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data.Schema	42
7.2	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface.Schema	44
7.3	http://schemas.microsoft.com/2003/10/Serialization/Arrays.Schema	45
7.4	http://schemas.microsoft.com/2003/10/Serialization/Schema	45
7.5	http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Schema	46
8	Appendix C: Product Behavior	48
9	Change Tracking.....	49
10	Index.....	51

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 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

broadcast session: A sharing session initiated by a presenter that is used for sharing the presenter's view of a document with one or more attendees.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [\[SSL3\]](#) and [\[RFC5246\]](#).

presentation slide: A slide that contains the content that can be displayed during a slide show. A presentation slide can derive formatting and content from a main master slide or a title master slide.

slide show: A delivery of a sequence of presentation slides, typically to an audience.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [\[SOAP1.2-1/2003\]](#).

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

SOAP fault: A container for error and status information within a SOAP message. See [\[SOAP1.2-1/2007\]](#) section 5.4 for more information.

time code: A digital signal applied to a stream. The signal assigns a number to every frame of video, representing hours, minutes, seconds, and frames.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

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.

[MS-OBPRS] Microsoft Corporation, "[Office Broadcast Presentation Service](#)".

[MS-WOPI] Microsoft Corporation, "[Web Application Open Platform Interface Protocol](#)".

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

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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kaktivaya, 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., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

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

1.2.2 Informative References

None.

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 **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:

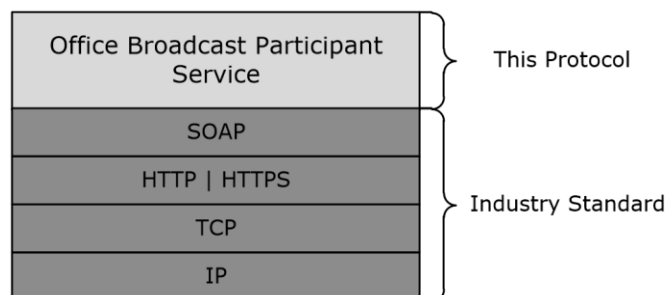


Figure 1: This protocol in relation to other protocols

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 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**.

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.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/	
tns1	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data	
tns2	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface	
tns3	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
tns4	http://schemas.microsoft.com/2003/10/Serialization/	
tns5	http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Imports	
wsaw	http://www.w3.org/2006/05/addressing/wsdl	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

Prefi x	Namespace URI	Reference
		[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 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="tns1: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="tns1: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="tns1:AppType"/>
    <xs:element minOccurs="0" name="BroadcastAppCapability"
type="tns1:BroadcastAppCapability"/>
    <xs:element minOccurs="0" name="BroadcastDataKey" type="tns1:BroadcastDataKey"/>
    <xs:element minOccurs="0" name="BroadcastState" type="tns1:BroadcastState"/>
    <xs:element minOccurs="0" name="PPTMediaPlaybackState"
type="tns1:PPTMediaPlaybackState"/>
    <xs:element minOccurs="0" name="PPTSlideShowState" type="tns1: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="tns1: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="tns1:ArrayOfPPTBroadcastAnimationStepData"/>
    <xs:element minOccurs="0" name="MediaStateDataList" nillable="true"
      type="tns1:ArrayOfPPTBroadcastMediaStateData"/>
    <xs:element minOccurs="0" name="PPTSlideShowState" type="tns1: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="tns2:ClientActions"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Type" type="tns2: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="tns2: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.
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.

Value	Meaning
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>
```

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>
```

```

    </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.
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.

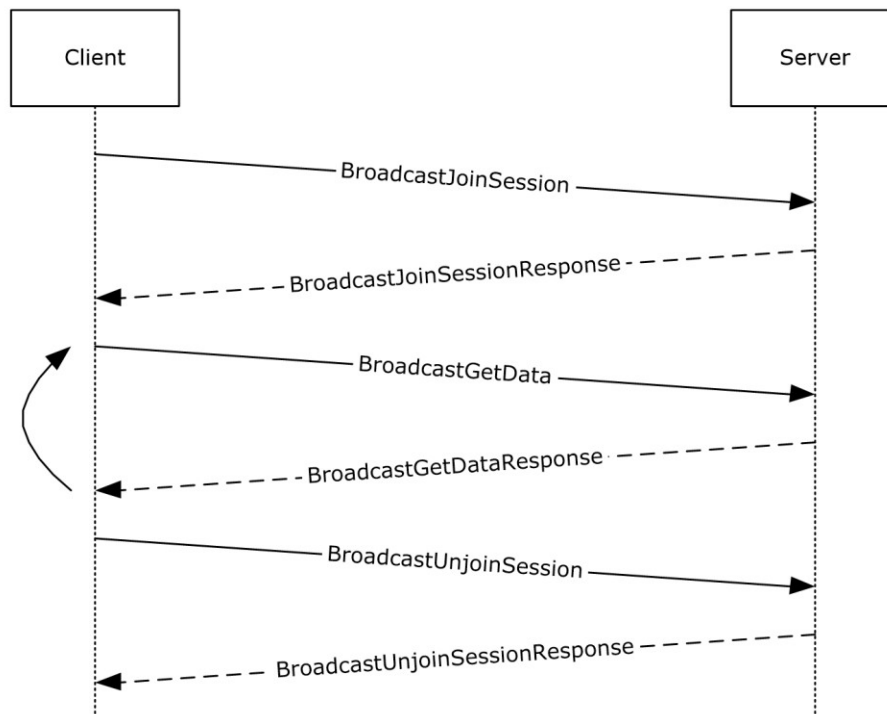


Figure 2: Office Broadcast Participant Service high-level sequence diagram for participant clients

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

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.

The following is the WSDL port type specification of the **BroadcastGetData WSDL operation**.

```
<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="tns: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="tns: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:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data" minOccurs="0" name="user" nillable="true" type="tns1: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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastGetDataResult" nillable="true"
type="tns2: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.

- 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>
```

</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.

The following is the WSDL port type specification of the **BroadcastJoinSession** WSDL operation.

```
<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="tns: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="tns: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

Message	Description
	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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastJoinSessionResult" nillable="true"
        type="tns2: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.

The following is the WSDL port type specification of the **BroadcastUnjoinSession** WSDL operation.

```
<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="tns: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="tns: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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"
        minOccurs="0" name="BroadcastUnjoinSessionResult" nillable="true"
        type="tns2: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.

The following is the WSDL port type specification of the **BroadcastPing** WSDL operation.

```
<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="tns: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="tns: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 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/BroadcastParticipantService/">
      <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/BroadcastParticipantService/">
      <BroadcastJoinSessionResult
        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: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&access_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/BroadcastParticipantService/">
      <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}], "PPTSlideShowStat
e":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" encoding="UTF-8"?>
<wsdl:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema
      xmlns:tns5="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Imports"
      targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Imports">
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data"/>
      <xs:import
        namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
      <xs:import
        namespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/" />
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IParticipantService">
    <wsdl:operation name="BroadcastJoinSession">
      <wsdl:input
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastJoinSession"
        message="tns:IParticipantService_BroadcastJoinSession_InputMessage" />
      <wsdl:output
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastJoinSessionResponse"
        message="tns:IParticipantService BroadcastJoinSession OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="BroadcastUnjoinSession">
      <wsdl:input
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastUnjoinSession"
        message="tns:IParticipantService BroadcastUnjoinSession InputMessage" />
      <wsdl:output
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastUnjoinSessionResponse"
        message="tns:IParticipantService_BroadcastUnjoinSession_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="BroadcastGetData">
      <wsdl:input
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastGetData"
        message="tns:IParticipantService_BroadcastGetData_InputMessage" />
      <wsdl:output
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastGetDataResponse"
        message="tns:IParticipantService_BroadcastGetData_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="BroadcastPing">
      <wsdl:input
        wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/IParticipantService/BroadcastPing"
        message="tns:IParticipantService_BroadcastPing_InputMessage" />
    </wsdl:operation>
  </wsdl:portType>

```

```

        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastPartici
pantService/IParticipantService/BroadcastPingResponse"
message="tns:IParticipantService_BroadcastPing_OutputMessage"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="DefaultBinding_IParticipantService" type="tns: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:message name="IParticipantService BroadcastGetData InputMessage">
        <wsdl:part name="parameters" element="tns:BroadcastGetData"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService BroadcastGetData OutputMessage">
        <wsdl:part name="parameters" element="tns:BroadcastGetDataResponse"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastJoinSession_InputMessage">
        <wsdl:part name="parameters" element="tns:BroadcastJoinSession"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService BroadcastJoinSession OutputMessage">
        <wsdl:part name="parameters" element="tns:BroadcastJoinSessionResponse"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastPing_InputMessage">
        <wsdl:part name="parameters" element="tns:BroadcastPing"/>
    </wsdl:message>
    <wsdl:message name="IParticipantService_BroadcastPing_OutputMessage">

```



```
<wsdl:part name="parameters" element="tns:BroadcastPingResponse"/>
</wsdl:message>
<wsdl:message name="IParticipantService_BroadcastUnjoinSession_InputMessage">
  <wsdl:part name="parameters" element="tns:BroadcastUnjoinSession"/>
</wsdl:message>
<wsdl:message name="IParticipantService_BroadcastUnjoinSession_OutputMessage">
  <wsdl:part name="parameters" element="tns:BroadcastUnjoinSessionResponse"/>
</wsdl:message>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

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

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:tns1="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="tns1:BroadcastUser"/>
  <xs:complexType name="PPTStateData">
    <xs:sequence>
      <xs:element minOccurs="0" name="AnimationStepDataList" nillable="true"
        type="tns1:ArrayOfPPTBroadcastAnimationStepData"/>
      <xs:element minOccurs="0" name="MediaStateDataList" nillable="true"
        type="tns1:ArrayOfPPTBroadcastMediaStateData"/>
      <xs:element minOccurs="0" name="PPTSlideShowState" type="tns1: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="tns1:PPTStateData"/>
  <xs:complexType name="ArrayOfPPTBroadcastAnimationStepData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="PPTBroadcastAnimationStepData"
        nillable="true" type="tns1:PPTBroadcastAnimationStepData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ArrayOfPPTBroadcastAnimationStepData" nillable="true"
    type="tns1:ArrayOfPPTBroadcastAnimationStepData"/>
  <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="tns1:PPTBroadcastAnimationStepData"/>
</xs:schema>
```

```

<xs:complexType name="ArrayOfPPTBroadcastMediaStateData">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="PPTBroadcastMediaStateData"
nillable="true" type="tns1:PPTBroadcastMediaStateData"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfPPTBroadcastMediaStateData" nillable="true"
type="tns1: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="tns1:PPTMediaPlaybackState"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="PPTBroadcastMediaStateData" nillable="true"
type="tns1: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="tns1: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="tns1: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="tns1:WordStateData"/>
<xs:complexType name="EnumTypes">
  <xs:sequence>
    <xs:element minOccurs="0" name="AppType" type="tns1:AppType"/>
    <xs:element minOccurs="0" name="BroadcastAppCapability"
type="tns1:BroadcastAppCapability"/>
    <xs:element minOccurs="0" name="BroadcastDataKey" type="tns1:BroadcastDataKey"/>
    <xs:element minOccurs="0" name="BroadcastState" type="tns1:BroadcastState"/>
    <xs:element minOccurs="0" name="PPTMediaPlaybackState"
type="tns1:PPTMediaPlaybackState"/>
    <xs:element minOccurs="0" name="PPTSlideShowState" type="tns1:PPTSlideShowState"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="EnumTypes" nillable="true" type="tns1: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="tns1: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:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="MediaExtensions"/>
        <xs:enumeration value="MaxMediaSize"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="BroadcastAppCapability" nillable="true"
type="tns1: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="tns1: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="tns1:BroadcastState"/>
</xs:schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:tns2="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="tns2:ServiceError"/>
            <xs:element minOccurs="0" name="Result" nillable="true" type="xs:anyType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ServiceResult" nillable="true" type="tns2: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="tns2:ClientActions"/>
            <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
            <xs:element minOccurs="0" name="Type" type="tns2:ServiceErrorType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ServiceError" nillable="true" type="tns2: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:simpleType>
<xs:element name="ClientActions" nillable="true" type="tns2: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="tns2:ServiceErrorType"/>
</xs:schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns3="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 name="ArrayOfKeyValueOfstringstring" nillable="true"
    type="tns3:ArrayOfKeyValueOfstringstring"/>
</xs:schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns4="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="tns4:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>

```

```

</xs:simpleType>
<xs:element name="duration" nillable="true" type="tns4:duration"/>
<xs:simpleType name="duration">
  <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>
<xs:element name="guid" nillable="true" type="tns4: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" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"
          minOccurs="0" name="BroadcastJoinSessionResult" nillable="true"
          type="tns2: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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"
          minOccurs="0" name="BroadcastUnjoinSessionResult" nillable="true"
          type="tns2:ServiceResult"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="BroadcastGetData">
    <xs:complexType>
      <xs:sequence>

```

```

        <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data" minOccurs="0" name="user" nillable="true" type="tns1: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:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface" minOccurs="0" name="BroadcastGetDataResult" nillable="true"
type="tns2: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:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data"/>
<xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface"/>
</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
- Microsoft OneNote 2016
- Microsoft Skype for Business Server 2015

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 this document since the last release. 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- 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 dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
8 Appendix C: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

10 Index

A

Abstract data model
 [server](#) 24
[Applicability](#) 9
[AppType simple type](#) 17
[ArrayOfKeyValueOfstringstring complex type](#) 12
[ArrayOfPPTBroadcastAnimationStepData complex type](#) 12
[ArrayOfPPTBroadcastMediaStateData complex type](#) 12
[Attendee Client example](#) 35
[Attribute groups](#) 22
[Attributes](#) 22

B

[BroadcastAppCapability simple type](#) 17
[BroadcastDataKey simple type](#) 18
[BroadcastState simple type](#) 19

C

[Capability negotiation](#) 9
[Change tracking](#) 49
[char simple type](#) 19
[ClientActions simple type](#) 19
[Complex types](#) 11
 [ArrayOfKeyValueOfstringstring](#) 12
 [ArrayOfPPTBroadcastAnimationStepData](#) 12
 [ArrayOfPPTBroadcastMediaStateData](#) 12
 [EnumTypes](#) 13
 [PPTBroadcastAnimationStepData](#) 13
 [PPTBroadcastMediaStateData](#) 14
 [PPTStateData](#) 14
 [ServiceError](#) 15
 [ServiceResult](#) 15
 [WordStateData](#) 16

D

Data model - abstract
 [server](#) 24
[duration simple type](#) 20

E

[EnumTypes complex type](#) 13
Events
 [local - server](#) 34
 [timer - server](#) 34
Examples
 [Attendee Client](#) 35
 [Overview](#) 35

F

[Fields - vendor-extensible](#) 9
[Full WSDL](#) 39
[Full XML schema](#) 42

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.DataSchema> 42

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.InterfaceSchema> 44

<http://schemas.microsoft.com/2003/10/Serialization/Schema> 45

<http://schemas.microsoft.com/2003/10/Serialization/ArraysSchema> 45

<http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Schema> 46

G

[Glossary](#) 6
[Groups](#) 22
[guid simple type](#) 20

I

[Implementer - security considerations](#) 38
[Index of security parameters](#) 38
[Informative references](#) 8
Initialization
 [server](#) 24
[Introduction](#) 6

L

Local events
 [server](#) 34

M

Message processing
 [server](#) 24
Messages
 [AppType simple type](#) 17
 [ArrayOfKeyValueOfstringstring complex type](#) 12
 [ArrayOfPPTBroadcastAnimationStepData complex type](#) 12
 [ArrayOfPPTBroadcastMediaStateData complex type](#) 12
 [attribute groups](#) 22
 [attributes](#) 22
 [BroadcastAppCapability simple type](#) 17
 [BroadcastDataKey simple type](#) 18
 [BroadcastState simple type](#) 19
 [char simple type](#) 19
 [ClientActions simple type](#) 19
 [complex types](#) 11
 [duration simple type](#) 20
 [elements](#) 11
 [enumerated](#) 11

EnumTypes complex type	13	BroadcastUnjoinSession operation	30
groups	22	initialization	24
guid simple type	20	local events	34
namespaces	10	message processing	24
PPTBroadcastAnimationStepData complex type	13	sequencing rules	24
PPTBroadcastMediaStateData complex type	14	timer events	34
PPTMediaPlaybackState simple type	20	timers	24
PPTSlideShowState simple type	21	ServiceError complex type	15
PPTStateData complex type	14	ServiceErrorType simple type	21
ServiceError complex type	15	ServiceResult complex type	15
ServiceErrorType simple type	21	Simple types	16
ServiceResult complex type	15	AppType	17
simple types	16	BroadcastAppCapability	17
syntax	10	BroadcastDataKey	18
transport	10	BroadcastState	19
WordStateData complex type	16	char	19
N		ClientActions	19
Namespaces	10	duration	20
Normative references	7	guid	20
O		PPTMediaPlaybackState	20
Operations		PPTSlideShowState	21
BroadcastGetData	25	ServiceErrorType	21
BroadcastJoinSession	28	Standards assignments	9
BroadcastPing	32	Syntax	
BroadcastUnjoinSession	30	messages - overview	10
Overview (synopsis)	8	T	
P		Timer events	
Parameters - security index	38	server	34
PPTBroadcastAnimationStepData complex type	13	Timers	
PPTBroadcastMediaStateData complex type	14	server	24
PPTMediaPlaybackState simple type	20	Tracking changes	49
PPTSlideShowState simple type	21	Transport	10
PPTStateData complex type	14	Types	
Preconditions	8	complex	11
Prerequisites	8	simple	16
Product behavior	48	V	
Protocol Details		Vendor-extensible fields	9
overview	23	Versioning	9
R		W	
References	7	WordStateData complex type	16
informative	8	WSDL	39
normative	7	X	
Relationship to other protocols	8	XML schema	42
S		http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Interface.Data Schema	42
Security		http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Broadcast.Pipe.Interface Schema	44
implementer considerations	38	http://schemas.microsoft.com/2003/10/Serialization/ Schema	45
parameter index	38	http://schemas.microsoft.com/2003/10/Serialization/Arrays Schema	45
Sequencing rules			
server	24		
Server			
abstract data model	24		
BroadcastGetData operation	25		
BroadcastJoinSession operation	28		
BroadcastPing operation	32		

<http://schemas.microsoft.com/office/Broadcast/Server/WebServices/BroadcastParticipantService/Schema> 46