

[MS-OBPRS]: Office Broadcast Presentation Service Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
01/20/2012	0.1	New	Released new document.
04/11/2012	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	0.2	Minor	Clarified the meaning of the technical content.

Table of Contents

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

3.1.4.1.1	Messages	24
3.1.4.1.1.1	BroadcastEndSessionSoapIn	24
3.1.4.1.1.2	BroadcastEndSessionSoapOut	24
3.1.4.1.2	Elements	24
3.1.4.1.2.1	BroadcastEndSession	24
3.1.4.1.2.2	BroadcastEndSessionResponse	25
3.1.4.1.3	Complex Types	25
3.1.4.1.4	Simple Types	25
3.1.4.1.5	Attributes	25
3.1.4.1.6	Groups	25
3.1.4.1.7	Attribute Groups	25
3.1.4.2	BroadcastGetAppCapabilities	25
3.1.4.2.1	Messages	26
3.1.4.2.1.1	BroadcastGetAppCapabilitiesSoapIn	26
3.1.4.2.1.2	BroadcastGetAppCapabilitiesSoapOut	26
3.1.4.2.2	Elements	26
3.1.4.2.2.1	BroadcastGetAppCapabilities	26
3.1.4.2.2.2	BroadcastGetAppCapabilitiesResponse	26
3.1.4.2.3	Complex Types	27
3.1.4.2.4	Simple Types	27
3.1.4.2.5	Attributes	27
3.1.4.2.6	Groups	27
3.1.4.2.7	Attribute Groups	28
3.1.4.3	BroadcastPutData	28
3.1.4.3.1	Messages	28
3.1.4.3.1.1	BroadcastPutDataSoapIn	28
3.1.4.3.1.2	BroadcastPutDataSoapOut	28
3.1.4.3.2	Elements	28
3.1.4.3.2.1	BroadcastPutData	29
3.1.4.3.2.2	BroadcastPutDataResponse	30
3.1.4.3.3	Complex Types	30
3.1.4.3.3.1	BroadcastDictionaryType	30
3.1.4.3.3.2	ItemType	30
3.1.4.3.4	Simple Types	31
3.1.4.3.5	Attributes	31
3.1.4.3.6	Groups	31
3.1.4.3.7	Attribute Groups	31
3.1.4.4	BroadcastStartSession	31
3.1.4.4.1	Messages	31
3.1.4.4.1.1	BroadcastStartSessionSoapIn	32
3.1.4.4.1.2	BroadcastStartSessionSoapOut	32
3.1.4.4.2	Elements	32
3.1.4.4.2.1	BroadcastStartSession	32
3.1.4.4.2.2	BroadcastStartSessionResponse	32
3.1.4.4.3	Complex Types	33
3.1.4.4.4	Simple Types	33
3.1.4.4.4.1	AppType	33
3.1.4.4.5	Attributes	33
3.1.4.4.6	Groups	34
3.1.4.4.7	Attribute Groups	34
3.1.4.5	BroadcastPing	34
3.1.4.5.1	Messages	34
3.1.4.5.1.1	BroadcastPingSoapIn	34

3.1.4.5.1.2 BroadcastPingSoapOut	34
3.1.4.5.2 Elements.....	34
3.1.4.5.2.1 BroadcastPing	35
3.1.4.5.2.2 BroadcastPingResponse.....	35
3.1.4.5.3 Complex Types	35
3.1.4.5.4 Simple Types.....	35
3.1.4.5.5 Attributes.....	35
3.1.4.5.6 Groups.....	35
3.1.4.5.7 Attribute Groups	35
3.1.5 Timer Events	35
3.1.6 Other Local Events	36
4 Protocol Examples.....	37
4.1 Presenter Client Example	37
5 Security.....	41
5.1 Security Considerations for Implementers.....	41
5.2 Index of Security Parameters	41
6 Appendix A: Full WSDL.....	42
7 Appendix B: Product Behavior	50
8 Change Tracking.....	51
9 Index	53

1 Introduction

The Office Broadcast Presentation Service Protocol enables a protocol client to create and update information about the state of a document being shared on a protocol server.

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

1.1 Glossary

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

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

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

broadcast session
presentation slide
Simple Object Access Protocol (SOAP)
slide show
SOAP action
SOAP body
SOAP fault
time code
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
WSDL message
WSDL operation
XML namespace
XML namespace prefix
XML schema

The following terms are specific to this document:

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

1.2 References

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

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

- [MS-OBPAS] Microsoft Corporation, "[Office Broadcast Participant Service Specification](#)".
- [MS-WOPI] Microsoft Corporation, "[Web Application Open Platform Interface Protocol Specification](#)".
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>
- [RFC4627] Crockford, D., "The application/json Media Type for Javascript Object Notation (JSON)", RFC 4627, July 2006, <http://www.ietf.org/rfc/rfc4627.txt>
- [SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624>
- [SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, <http://www.w3.org/TR/2003/REC-soap12-part2-20030624>
- [WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>
- [XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>
- [XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

- [MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".
- [MS-OFGLGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.ietf.org/rfc/rfc2818.txt>

1.3 Overview

When meeting remotely, it is useful to share the current view of a document so others might 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 begin or end a document **broadcast session**, and to store data about the state of a broadcast session on the protocol server.

1.4 Relationship to Other Protocols

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

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

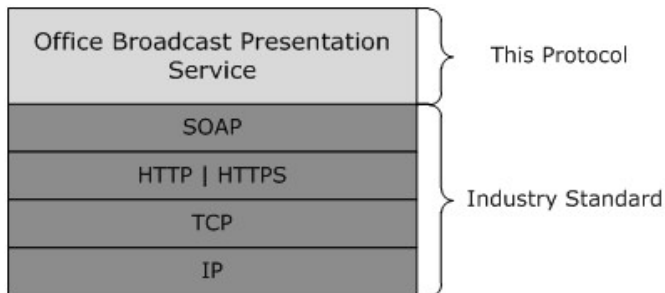


Figure 1: This protocol in relation to other protocols

This protocol works in conjunction with the Office Broadcast Participant Service Protocol as specified in [\[MS-OBPAS\]](#) and uses the Web Application Open Platform Interface Protocol for file retrieval as specified 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/Present_2_0.asmx" to the URL of the site, for example: http://www.example.com/m/Present_2_0.asmx.

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 begin and end broadcast sessions on the protocol server and to store and update broadcast session information on the protocol server.

1.7 Versioning and Capability Negotiation

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

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might be less restrictive than the processing rules imposed by the protocol. The **WSDL** in this specification matches the WSDL that shipped with the product and provides a base description of the schema. The text that introduces the WSDL specifies additional restrictions 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 protocol 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 used by this protocol. The syntax of the definitions uses **XML schema** as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language (WSDL) as defined 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
http	http://schemas.xmlsoap.org/wsdl/http/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/server/broadcast/2010/main	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

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
AppServerInfo	A complex type that contains settings that dictate the behavior of the protocol server.
ArrayOfPPTBroadcastAnimationStepData	A complex type that specifies a list of PPTBroadcastAnimationStepData (section 2.2.4.6) 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 field.
ArrayOfPPTBroadcastMediaStateData	A complex type that specifies a list of PPTBroadcastMediaStateData (section 2.2.4.7) 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.
BroadcastUser	A complex type that specifies a user of a broadcast session.
EnumTypes	Reserved. 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 by 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 Word document.

2.2.4.1 AppServerInfo

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

A complex type that contains settings that dictate the behavior of the protocol server.

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

```

    <xs:element minOccurs="0" maxOccurs="1" name="AppCapabilities"
type="tns:BroadcastDictionaryType"/>
  </xs:sequence>
</xs:complexType>

```

AppCapabilities: A **BroadcastDictionaryType** (section [3.1.4.3.3.1](#)) element that contains server-related settings. This element **MUST** be present.

2.2.4.2 ArrayOfPPTBroadcastAnimationStepData

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

A complex type that specifies a list of **PPTBroadcastAnimationStepData** (section [2.2.4.6](#)) 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** field.

```

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

```

PPTBroadcastAnimationStepData: Specifies a **PPTBroadcastAnimationStepData** (section [2.2.4.6](#)) element. The element **MUST** be present.

2.2.4.3 ArrayOfPPTBroadcastMediaStateData

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

A complex type that specifies a list of **PPTBroadcastMediaStateData** (section [2.2.4.7](#)) elements. Each element in the list specifies a state for a multimedia object. Each entry in the list **MUST** have a unique value in its **MediaId** field.

```

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

```

PPTBroadcastMediaStateData: Each element specifies a **PPTBroadcastMediaStateData** (section [2.2.4.7](#)) element. The element **MUST** be present.

2.2.4.4 BroadcastUser

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

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" maxOccurs="1" name="SessionId" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserToken" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

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

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

2.2.4.5 EnumTypes

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

Reserved. **MUST** be ignored.

```

<xs:complexType name="EnumTypes" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="BroadcastDataKey"
type="tns:BroadcastDataKey"/>
    <xs:element minOccurs="1" maxOccurs="1" name="BroadcastAppCapability"
type="tns:BroadcastAppCapability"/>
    <xs:element minOccurs="1" maxOccurs="1" name="AppType" type="tns:AppType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="BroadcastState" type="tns:BroadcastState"/>
    <xs:element minOccurs="1" maxOccurs="1" name="PPTSlideShowState"
type="tns:PPTSlideShowState"/>
    <xs:element minOccurs="1" maxOccurs="1" name="PPTMediaPlaybackState"
type="tns:PPTMediaPlaybackState"/>
  </xs:sequence>
</xs:complexType>

```

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

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

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

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

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

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

2.2.4.6 PPTBroadcastAnimationStepData

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

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" maxOccurs="1" name="TimelineId" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Step" type="xs:int"/>
  </xs:sequence>

```

```
</xs:complexType>
```

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

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.

2.2.4.7 PPTBroadcastMediaStateData

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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" maxOccurs="1" name="MediaId" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="State" type="tns:PPTMediaPlaybackState"/>
    <xs:element minOccurs="1" maxOccurs="1" name="AtPosition" type="xs:double"/>
  </xs:sequence>
</xs:complexType>
```

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 2.2.5.5) element that specifies the playback state of the multimedia object identified by the **MediaId** field. This element **MUST** be present.

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.

2.2.4.8 PPTStateData

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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="1" maxOccurs="1" name="SlideId" type="xs:unsignedInt"/>
    <xs:element minOccurs="1" maxOccurs="1" name="SlideIndex" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" maxOccurs="1" name="AnimationStepDataList"
      type="tns:ArrayOfPPTBroadcastAnimationStepData"/>
    <xs:element minOccurs="0" maxOccurs="1" name="MediaStateDataList"
      type="tns:ArrayOfPPTBroadcastMediaStateData"/>
    <xs:element minOccurs="1" maxOccurs="1" name="PPTSlideShowState"
      type="tns:PPTSlideShowState"/>
  </xs:sequence>
</xs:complexType>
```

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.

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 2.2.5.6) element that specifies the current state of the **slide show**. This element MUST be present.

2.2.4.9 ServiceError

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

A complex type that specifies error information returned by 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" maxOccurs="1" name="Message" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Title" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ServiceErrorType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="RecommendedActions"
      type="tns:ClientActions"/>
  </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.

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 2.2.5.7) element that specifies the error type. This element MUST be present.

RecommendedActions: Reserved and MUST be ignored.

2.2.4.10 ServiceResult

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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" maxOccurs="1" name="Result"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Error" type="tns:ServiceError"/>
  </xs:sequence>
</xs:complexType>
```

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

Error: An optional **ServiceError** (section [2.2.4.9](#)) element that specifies an error result of a protocol message response. This element **MUST NOT** be present if the **Result** element is present.

2.2.4.11 WordStateData

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

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

```
<xs:complexType name="WordStateData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Page" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Offset" type="xs:double"/>
  </xs:sequence>
</xs:complexType>
```

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.

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.

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
BroadcastAppCapability	A simple type that specifies an enumeration of a set of capabilities returned by the protocol server to the protocol client.
BroadcastDataKey	A simple type that specifies an enumeration of a set of keys to data stored in the protocol server for a given broadcast session.
BroadcastState	A simple type that specifies an enumeration of all the possible broadcast session states.
ClientActions	A simple type that is reserved and MUST be ignored.
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 BroadcastAppCapability

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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

```
<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 **BroadcastAppCapability**.

Value	Meaning
SessionTimeout	This capability specifies the maximum duration in seconds for a broadcast session after which the session is ended.
SessionIdleTimeOut	This capability specifies the maximum duration in seconds after which a broadcast session is ended if there has been no presenter input.
SupportVideo	This capability specifies whether the protocol server supports video content in the broadcast session.
SupportAudio	This capability specifies whether the protocol server supports audio content in the broadcast session.
SupportNotes	This capability specifies whether the protocol server supports note-taking through OneNote in the broadcast session.
MediaExtensions	This capability specifies the supported file name extensions for multimedia content.
MaxMediaSize	This capability specifies the maximum size in bytes of an individual multimedia object that is rendered on the protocol server.

2.2.5.2 BroadcastDataKey

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

A simple type that specifies an enumeration of a set of keys to data stored in the protocol server for a given broadcast 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:restriction>
</xs:simpleType>
```



```

    <xs:enumeration value="AppSpecificStateData"/>
    <xs:enumeration value="NotesUrl"/>
    <xs:enumeration value="DataVersion"/>
  </xs:restriction>
</xs:simpleType>

```

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

Value	Meaning
AppType	This key is used to specify an application type in the key field of an ItemType (section 3.1.4.3.3.2).
SequenceNumber	This key is used to specify a monotonically increasing sequence number in the key field of an ItemType (section 3.1.4.3.3.2).
FileVersion	This key is used to specify a monotonically increasing file version number in the key field of an ItemType (section 3.1.4.3.3.2).
OriginalFileName	This key is used to specify the file name of the document in the key field of an ItemType (section 3.1.4.3.3.2).
BroadcastState	This key is used to specify the state of a broadcast session in the key field of an ItemType (section 3.1.4.3.3.2).
AppSpecificStateData	This key is used to specify application specific data in the key field of an ItemType (section 3.1.4.3.3.2).
NotesUrl	This key is used to specify the URL of the OneNote page linked to the document in the key field of an ItemType (section 3.1.4.3.3.2).
DataVersion	This key is used to specify a data version field in the key field of an ItemType (section 3.1.4.3.3.2).

2.2.5.3 BroadcastState

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

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 **BroadcastState**.

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

Value	Meaning
BroadcastEnded	The broadcast session has ended.

2.2.5.4 ClientActions

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

A simple type that is reserved and MUST be ignored.

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

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

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.

2.2.5.5 PPTMediaPlaybackState

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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 **PPTMediaPlaybackState**.

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.

2.2.5.6 PPTSlideShowState

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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 **PPTSlideShowState**.

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.

2.2.5.7 ServiceErrorType

Namespace: http://schemas.microsoft.com/server/broadcast/2010/main

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 **ServiceErrorType**.

Value	Meaning
UnknownError	The protocol server encountered an unknown error.
ApplicationError	The protocol server encountered an application error.
Timeout	The protocol server encountered an application timeout.
ServiceBusy	The protocol server is busy.
SessionFull	The protocol server 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 presenter client protocol.

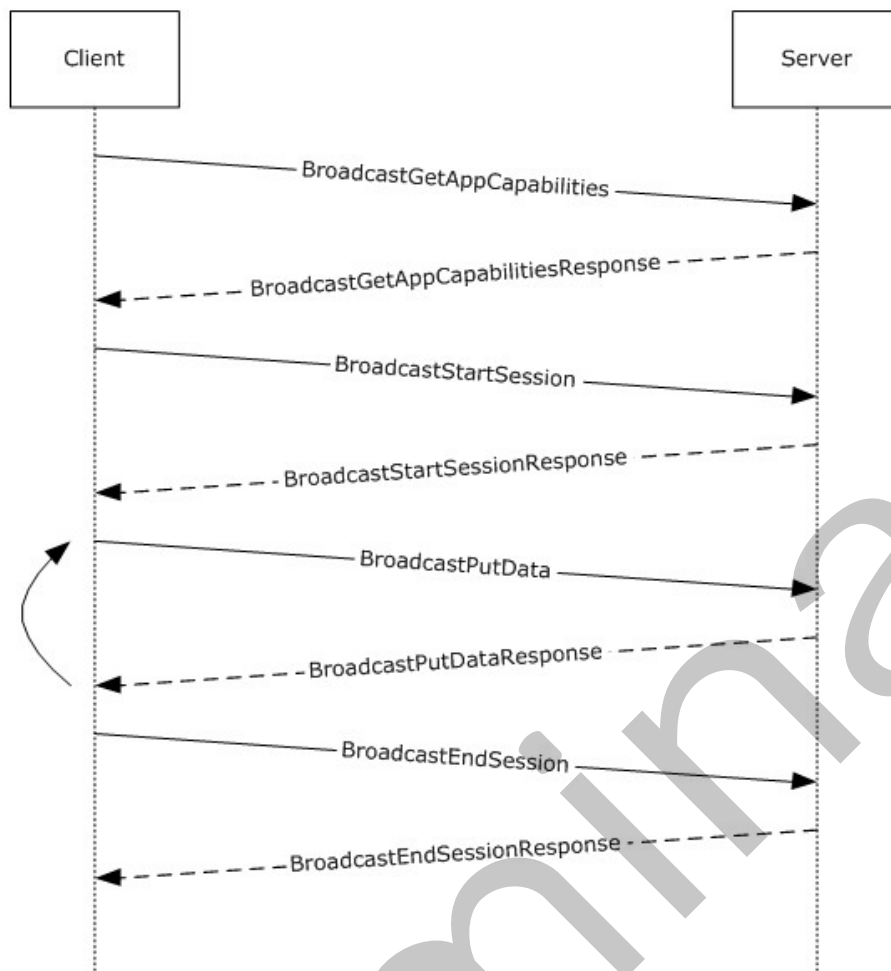


Figure 2: Office Broadcast Presentation Service high-level sequence diagram for presenter clients

First, a protocol client acting as broadcast presenter sends a **BroadcastGetAppCapabilities** (section 3.1.4.2) message. The server responds with a **BroadcastGetAppCapabilitiesResponse** message that contains the capabilities supported by the server. Next, the protocol client sends a **BroadcastStartSession** (section 3.1.4.4) message and the protocol server responds with a **BroadcastStartSessionResponse** message indicating the result of the operation. Next, the protocol client sends one or more **BroadcastPutData** (section 3.1.4.3) messages containing the current state of the broadcast, and the server sends a **BroadcastPutDataResponse** message to acknowledge the request. When the broadcast is finished, the protocol client then sends a **BroadcastEndSession** (section 3.1.4.1) message and the protocol server responds with a **BroadcastEndSessionResponse** message to acknowledge the request.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

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

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

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

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

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

Section [3.1](#) specifies the sequencing of the protocol messages and how they relate to each other. The following sections specify the details of each individual message.

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

Operation	Description
BroadcastEndSession	The BroadcastEndSession operation is used by the protocol client to end a broadcast session on the protocol server.
BroadcastGetAppCapabilities	The BroadcastGetAppCapabilities operation is used by the protocol client to retrieve session-independent settings from the protocol server.
BroadcastPing	The BroadcastPing operation is used by a protocol client to check if the protocol server is available.
BroadcastPutData	The BroadcastPutData operation is used by the protocol client to modify the current state of the broadcast session on the protocol server.
BroadcastStartSession	The BroadcastStartSession operation is used by the protocol client to begin a broadcast session on the protocol server.

3.1.4.1 BroadcastEndSession

The **BroadcastEndSession** operation is used by the protocol client to end a broadcast session on the protocol server.

```
<wsdl:operation name="BroadcastEndSession" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:BroadcastEndSessionSoapIn"/>
  <wsdl:output message="tns:BroadcastEndSessionSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastEndSessionSoapIn** request message, as specified in section [3.1.4.1.1.1](#), and the protocol server MUST respond with a **BroadcastEndSessionSoapOut** response message, as specified in section [3.1.4.1.1.2](#).

3.1.4.1.1 Messages

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

Message	Description
BroadcastEndSessionSoapIn	The request WSDL message for the BroadcastEndSession WSDL operation .
BroadcastEndSessionSoapOut	The response WSDL message for the BroadcastEndSession WSDL operation .

3.1.4.1.1.1 BroadcastEndSessionSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastEndSession
```

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

3.1.4.1.1.2 BroadcastEndSessionSoapOut

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

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

3.1.4.1.2 Elements

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

Element	Description
BroadcastEndSession	The input data for the BroadcastEndSession WSDL operation .
BroadcastEndSessionResponse	The result data for the BroadcastEndSession WSDL operation .

3.1.4.1.2.1 BroadcastEndSession

The input data for the **BroadcastEndSession WSDL operation**.

```
<xs:element name="BroadcastEndSession" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="user" type="tns:BroadcastUser"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

user: A **BroadcastUser** element (section [2.2.4.4](#)) that is obtained through a **BroadcastStartSession** operation (section [3.1.4.4](#)). This element **MUST** be present.

3.1.4.1.2.2 BroadcastEndSessionResponse

The result data for the **BroadcastEndSession** WSDL operation.

```
<xs:element name="BroadcastEndSessionResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastEndSessionResult"
        type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastEndSessionResult: A **ServiceResult** (section [2.2.4.10](#)) 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.

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 BroadcastGetAppCapabilities

The **BroadcastGetAppCapabilities** operation is used by the protocol client to retrieve session-independent settings from the protocol server.

```
<wsdl:operation name="BroadcastGetAppCapabilities"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:BroadcastGetAppCapabilitiesSoapIn"/>
  <wsdl:output message="tns:BroadcastGetAppCapabilitiesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastGetAppCapabilitiesSoapIn** request message, as specified in section [3.1.4.2.1.1](#), and the protocol server **MUST** respond with a **BroadcastGetAppCapabilitiesSoapOut** response message, as specified in section [3.1.4.2.1.2](#).

3.1.4.2.1 Messages

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

Message	Description
BroadcastGetAppCapabilitiesSoapIn	The request WSDL message for the BroadcastGetAppCapabilities WSDL operation.
BroadcastGetAppCapabilitiesSoapOut	The response WSDL message for the BroadcastGetAppCapabilities WSDL operation.

3.1.4.2.1.1 BroadcastGetAppCapabilitiesSoapIn

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

The SOAP action value is:

```
http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastGetAppCapabilities
```

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

3.1.4.2.1.2 BroadcastGetAppCapabilitiesSoapOut

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

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

3.1.4.2.2 Elements

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

Element	Description
BroadcastGetAppCapabilities	The input data for the BroadcastGetAppCapabilities WSDL operation.
BroadcastGetAppCapabilitiesResponse	The result data for the BroadcastGetAppCapabilities WSDL operation.

3.1.4.2.2.1 BroadcastGetAppCapabilities

The input data for the **BroadcastGetAppCapabilities** WSDL operation.

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

3.1.4.2.2.2 BroadcastGetAppCapabilitiesResponse

The result data for the **BroadcastGetAppCapabilities** WSDL operation.

```

<xs:element name="BroadcastGetAppCapabilitiesResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastGetAppCapabilitiesResult"
        type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

BroadcastGetAppCapabilitiesResult: A **ServiceResult** (section [2.2.4.10](#)) that specifies the result of the operation. This element **MUST** be present. If successful, the **Result** child element of the **ServiceResult** **MUST** be an **AppServerInfo** (section [2.2.4.1](#)) element. Otherwise, the **Error** child element of the **ServiceResult** **MUST** be present.

The **AppServerInfo** contains a **BroadcastDictionaryType** (section [3.1.4.3.3.1](#)), which contains a list of **ItemType** (section [3.1.4.3.3.2](#)) elements.

The **key** field in an **ItemType** element **MUST** be a **BroadcastAppCapability** (section [2.2.5.1](#)) element.

The **value** field of an **ItemType** element **MUST** be an **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element, as specified in the following table.

Key	Value
SessionTimeout	MUST be a non-negative integer
SessionIdleTimeOut	MUST be a non-negative integer
SupportVideo	MUST be "true" or "false"
SupportAudio	MUST be "true" or "false"
SupportNotes	MUST be "true" or "false"
MediaExtensions	MUST be a string containing a comma-separated list of file name extensions
MaxMediaSize	MUST be a non-negative integer

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 BroadcastPutData

The **BroadcastPutData** operation is used by the protocol client to modify the current state of the broadcast session on the protocol server.

```
<wsdl:operation name="BroadcastPutData" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:BroadcastPutDataSoapIn"/>
  <wsdl:output message="tns:BroadcastPutDataSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastPutDataSoapIn** request message, as specified in section [3.1.4.3.1.1](#), and the protocol server MUST respond with a **BroadcastPutDataSoapOut** response message, as specified in section [3.1.4.3.1.2](#).

3.1.4.3.1 Messages

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

Message	Description
BroadcastPutDataSoapIn	The request WSDL message for the BroadcastPutData WSDL operation.
BroadcastPutDataSoapOut	The response WSDL message for the BroadcastPutData WSDL operation.

3.1.4.3.1.1 BroadcastPutDataSoapIn

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

The SOAP action value is:

```
http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPutData
```

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

3.1.4.3.1.2 BroadcastPutDataSoapOut

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

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

3.1.4.3.2 Elements

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

Element	Description
BroadcastPutData	The input data for the BroadcastPutData WSDL operation.

Element	Description
BroadcastPutDataResponse	The result data for the BroadcastPutData WSDL operation.

3.1.4.3.2.1 BroadcastPutData

The input data for the **BroadcastPutData** WSDL operation.

```
<xs:element name="BroadcastPutData" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="user" type="tns:BroadcastUser"/>
      <xs:element minOccurs="0" maxOccurs="1" name="data"
type="tns:BroadcastDictionaryType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

user: A **BroadcastUser** (section [2.2.4.4](#)) element that is obtained by making a **BroadcastStartSession** (section [3.1.4.4](#)) web method call. This element **MUST** be present.

data: A **BroadcastDictionaryType** (section [3.1.4.3.3.1](#)) element that specifies the current broadcast session on the protocol client. This element **MUST** be present.

The **BroadcastDictionaryType** contains a list of **ItemType** (section [3.1.4.3.3.2](#)) elements. The **key** in each **ItemType** element **MUST** be a **BroadcastDataKey** (section [2.2.5.2](#)) value of type **xs:string** [[XMLSCHEMA2](#)] section 3.2.1.

The **value** field in an **ItemType** element **MUST** conform to the following table.

Key	Value
AppType	MUST be an AppType (section 3.1.4.4.4.1) element.
SequenceNumber	MUST be an xs:string [XMLSCHEMA2] section 3.2.1 element. The numeric value MUST follow an xs:int (XMLSCHEMA2) section 3.3.17) element.
FileVersion	MUST be an xs:string [XMLSCHEMA2] section 3.2.1 element. The numeric value MUST follow 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 2.2.5.3) element.
AppSpecificStateData	MUST be an xs:string [XMLSCHEMA2] section 3.2.1 element. The AppSpecificStateData MUST have the following layout: MUST be a JavaScript Object Notation (JSON) [RFC4627] serialization of a PPTStateData (section 2.2.4.8) element if the AppType is PPT , or a JSON [RFC4627] serialization of a WordStateData (section 2.2.4.11) element if the AppType is Word .
NotesUrl	MUST be an xs:string (XMLSCHEMA2) section 3.2.1) element.
DataVersion	MUST be an xs:int (XMLSCHEMA2) section 3.3.17) element with a value of 2.

3.1.4.3.2 BroadcastPutDataResponse

The result data for the **BroadcastPutData** WSDL operation.

```
<xs:element name="BroadcastPutDataResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastPutDataResult"
type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

BroadcastPutDataResult: A **ServiceResult** (section [2.2.4.10](#)) 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

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

Complex type	Description
BroadcastDictionaryType	A complex type that specifies a list of ItemType (section 3.1.4.3.3.2) elements.
ItemType	A complex type that holds a key-value pair.

3.1.4.3.3.1 BroadcastDictionaryType

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

A complex type that specifies a list of **ItemType** (section [3.1.4.3.3.2](#)) elements.

```
<xs:complexType name="BroadcastDictionaryType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="item" type="tns:ItemType"/>
  </xs:sequence>
</xs:complexType>
```

item: Each element specifies an **ItemType** (section [3.1.4.3.3.2](#)) element. The element **MUST** be present.

3.1.4.3.3.2 ItemType

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

A complex type that holds a key-value pair.

```
<xs:complexType name="ItemType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element name="key" type="xs:string"/>
    <xs:element name="value" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```
</xs:sequence>
</xs:complexType>
```

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

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

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 BroadcastStartSession

The **BroadcastStartSession** operation is used by the protocol client to begin a broadcast session on the protocol server.

```
<wsdl:operation name="BroadcastStartSession" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:BroadcastStartSessionSoapIn"/>
  <wsdl:output message="tns:BroadcastStartSessionSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastStartSessionSoapIn** request message, as specified in section [3.1.4.4.1.1](#), and the protocol server **MUST** respond with a **BroadcastStartSessionSoapOut** 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
BroadcastStartSessionSoapIn	The request WSDL message for the BroadcastStartSession WSDL operation.
BroadcastStartSessionSoapOut	The response WSDL message for the BroadcastStartSession WSDL operation.

3.1.4.4.1.1 BroadcastStartSessionSoapIn

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

The SOAP action value is:

```
http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastStartSession
```

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

3.1.4.4.1.2 BroadcastStartSessionSoapOut

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

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

3.1.4.4.2 Elements

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

Element	Description
BroadcastStartSession	The input data for the BroadcastStartSession WSDL operation.
BroadcastStartSessionResponse	The result data for the BroadcastStartSession WSDL operation.

3.1.4.4.2.1 BroadcastStartSession

The input data for the **BroadcastStartSession** WSDL operation.

```
<xs:element name="BroadcastStartSession" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="query" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="appType" type="tns:AppType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

query: An **xs:string** element (see [XMLSCHEMA2](#) section 3.2.1) 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.

appType: An **AppType** (section [3.1.4.4.4.1](#)) element that specifies the type of the broadcast session.

3.1.4.4.2.2 BroadcastStartSessionResponse

The result data for the **BroadcastStartSession** WSDL operation.

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



```

<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="BroadcastStartSessionResult"
type="tns:ServiceResult"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

BroadcastStartSessionResult: A **ServiceResult** (section [2.2.4.10](#)) that specifies the result of the operation. This element **MUST** be present. The **Result** child element **MUST** be a **BroadcastUser** (section [2.2.4.4](#)) if the **Error** child element is not present.

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

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

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

3.1.4.4.4.1 AppType

Namespace: <http://schemas.microsoft.com/server/broadcast/2010/main>

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

```

<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 **AppType**.

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

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 BroadcastPing

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

```
<wsdl:operation name="BroadcastPing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:BroadcastPingSoapIn"/>
  <wsdl:output message="tns:BroadcastPingSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastPingSoapIn** request message, as specified in section [3.1.4.5.1.1](#), and the protocol server MUST respond with a **BroadcastPingSoapOut** response message, as specified in section [3.1.4.5.1.2](#).

3.1.4.5.1 Messages

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

Message	Description
BroadcastPingSoapIn	The request WSDL message for the BroadcastPing WSDL operation.
BroadcastPingSoapOut	The response WSDL message for the BroadcastPing WSDL operation.

3.1.4.5.1.1 BroadcastPingSoapIn

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

The SOAP action value is:

```
http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPing
```

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

3.1.4.5.1.2 BroadcastPingSoapOut

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

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

3.1.4.5.2 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.

3.1.4.5.2.1 BroadcastPing

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

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

3.1.4.5.2.2 BroadcastPingResponse

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="1" maxOccurs="1" 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.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

Preliminary

4 Protocol Examples

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

4.1 Presenter Client Example

The presenter protocol client begins by sending a request to the protocol server to fetch the settings stored on the protocol server. The following **BroadcastGetAppCapabilitiesSoapIn** message is sent to the protocol server:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastGetAppCapabilities
xmlns="http://schemas.microsoft.com/server/broadcast/2010/main" />
  </soap:Body>
</soap:Envelope>
```

The protocol server responds with a message to acknowledge the request and to provide the protocol client with the protocol server settings and their values. The following **BroadcastGetAppCapabilitiesSoapOut** message is sent to the presenter protocol client:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastGetAppCapabilitiesResponse
xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <BroadcastGetAppCapabilitiesResult>
        <Result xsi:type="AppServerInfo">
          <AppCapabilities>
            <item>
              <key>SessionTimeout</key>
              <value>43200</value>
            </item>
            <item>
              <key>SupportNotes</key>
              <value>true</value>
            </item>
            <item>
              <key>MediaExtensions</key>
              <value>3g2, 3gp, 3gpp, 3gp2, aac, adts, asf, asx, flv, m4a, m4b, m4v, mov, mp3, mp4, mpa, swf, wax, wm, wma, wmv,
              wmx, wpl, wvx</value>
            </item>
            <item>
              <key>MaxMediaSize</key>
              <value>10485760</value>
            </item>
          </AppCapabilities>
        </Result>
      </BroadcastGetAppCapabilitiesResult>
    </BroadcastGetAppCapabilitiesResponse>
  </soap:Body>
</soap:Envelope>
```

The presenter protocol client then sends a request to the protocol server to begin the broadcast. The following **BroadcastStartSessionSoapIn** message is sent to the protocol server:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastStartSession xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <query>WOPIsrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access_token=VAR</query>
      <appType>PPT</appType>
    </BroadcastStartSession>
  </soap:Body>
</soap:Envelope>
```

The protocol server responds with a message to acknowledge the request and to provide a user identifier and a session identifier to be used by the presenter protocol client for future requests. The following **BroadcastStartSessionSoapOut** message is sent to the presenter protocol client:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastStartSessionResponse
xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <BroadcastStartSessionResult>
        <Result xsi:type="BroadcastUser">
          <SessionId>WOPIsrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access_token=VAR</SessionId>
          <UserToken>e59840b8-d523-4477-b978-5369a237775a</UserToken>
        </Result>
      </BroadcastStartSessionResult>
    </BroadcastStartSessionResponse>
  </soap:Body>
</soap:Envelope>
```

Next, the presenter protocol client sends information about the current state of the broadcast. The following **BroadcastPutDataSoapIn** message is sent to the protocol server:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastPutData xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <user>
        <SessionId>WOPIsrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access_token=VAR</SessionId>
        <UserToken>e59840b8-d523-4477-b978-5369a237775a</UserToken>
      </user>
      <data>
        <item>
          <key>AppType</key>
          <value>PPT</value>
        </item>
        <item>
          <key>BroadcastState</key>
          <value>BroadcastStarted</value>
        </item>
      </data>
    </BroadcastPutData>
  </soap:Body>
</soap:Envelope>
```

```

    </item>
    <item>
      <key>SequenceNumber</key>
      <value>5</value>
    </item>
    <item>
      <key>FileVersion</key>
      <value>1</value>
    </item>
    <item>
      <key>OriginalFileName</key>
      <value>file.pptx</value>
    </item>
    <item>
      <key>NotesUrl</key>
      <value>http://url</value>
    </item>
    <item>
      <key>DataVersion</key>
      <value>2</value>
    </item>
    <item>
      <key>AppSpecificStateData</key>
      <value>{"SlideId":258,"SlideIndex":0,"AnimationStepDataList":[{"TimelineId":"timeline_1","Step":1}, {"TimelineId":"timeline_2","Step":2}], "MediaStateDataList":[{"MediaId":"306s4f1","State":1,"AtPosition":0.015}, {"MediaId":"55s21","State":0,"AtPosition":34.53}], "PPTSlideShowState":2}</value>
    </item>
  </data>
</BroadcastPutData>
</soap:Body>
</soap:Envelope>

```

The protocol server responds with a message to acknowledge the request. The following **BroadcastPutDataSoapOut** message is sent to the presenter protocol client:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastPutDataResponse
xmlns="http://schemas.microsoft.com/server/broadcast/2010/main"><BroadcastPutDataResult
/></BroadcastPutDataResponse>
  </soap:Body>
</soap:Envelope>

```

The presenter protocol client continues to send these **BroadcastPutDataSoapIn** messages until the broadcast is finished. At this point, the presenter protocol client sends a request to the protocol server to end the broadcast. The following **BroadcastEndSessionSoapIn** message is sent to the protocol server:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastEndSession xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <user>

```

```
<SessionId>WOPISrc=http://machinename/th/handler/wopi/files/anonymous~PPTTest.pptx&access_token=VAR</SessionId>
  <UserToken>e59840b8-d523-4477-b978-5369a237775a</UserToken>
</user>
</BroadcastEndSession>
</soap:Body>
</soap:Envelope>
```

The protocol server responds with a message to acknowledge the request. The following **BroadcastEndSessionSoapOut** message is sent to the presenter protocol client:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <BroadcastEndSessionResponse
xmlns="http://schemas.microsoft.com/server/broadcast/2010/main">
      <BroadcastEndSessionResult />
    </BroadcastEndSessionResponse>
  </soap:Body>
</soap:Envelope>
```


5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: Full WSDL

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

```
<?xml version="1.0"?>
<wsdl:definitions xmlns:tns="http://schemas.microsoft.com/server/broadcast/2010/main"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/server/broadcast/2010/main"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/server/broadcast/2010/main">
      <xs:element name="BroadcastStartSession">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="query" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="appType" type="tns:AppType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:simpleType name="AppType">
        <xs:restriction base="xs:string">
          <xs:enumeration value="PPT"/>
          <xs:enumeration value="Word"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:element name="BroadcastStartSessionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="BroadcastStartSessionResult"
type="tns:ServiceResult"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="ServiceResult">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Result"/>
          <xs:element minOccurs="0" maxOccurs="1" name="Error" type="tns:ServiceError"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ServiceError">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="Title" type="xs:string"/>
          <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ServiceErrorType"/>
          <xs:element minOccurs="1" maxOccurs="1" name="RecommendedActions"
type="tns:ClientActions"/>
        </xs:sequence>
      </xs:complexType>
      <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:schema>
  </wsdl:types>

```

```

    </xs:restriction>
  </xs:simpleType>
  <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:complexType name="AppServerInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="AppCapabilities"
type="tns:BroadcastDictionaryType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="WordStateData">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="Page" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="Offset" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PPTStateData">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="SlideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="1" maxOccurs="1" name="SlideIndex" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" maxOccurs="1" name="AnimationStepDataList"
type="tns:ArrayOfPPTBroadcastAnimationStepData"/>
      <xs:element minOccurs="0" maxOccurs="1" name="MediaStateDataList"
type="tns:ArrayOfPPTBroadcastMediaStateData"/>
      <xs:element minOccurs="1" maxOccurs="1" name="PPTSlideShowState"
type="tns:PPTSlideShowState"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfPPTBroadcastAnimationStepData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded"
name="PPTBroadcastAnimationStepData" nillable="true"
type="tns:PPTBroadcastAnimationStepData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PPTBroadcastAnimationStepData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="TimelineId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="Step" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfPPTBroadcastMediaStateData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="PPTBroadcastMediaStateData"
nillable="true" type="tns:PPTBroadcastMediaStateData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PPTBroadcastMediaStateData">

```

```

    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="MediaId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="State"
type="tns:PPTMediaPlaybackState"/>
      <xs:element minOccurs="1" maxOccurs="1" name="AtPosition" type="xs:double"/>
    </xs:sequence>
  </xs:complexType>
  <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: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:complexType name="EnumTypes">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="BroadcastDataKey"
type="tns:BroadcastDataKey"/>
      <xs:element minOccurs="1" maxOccurs="1" name="BroadcastAppCapability"
type="tns:BroadcastAppCapability"/>
      <xs:element minOccurs="1" maxOccurs="1" name="AppType" type="tns:AppType"/>
      <xs:element minOccurs="1" maxOccurs="1" name="BroadcastState"
type="tns:BroadcastState"/>
      <xs:element minOccurs="1" maxOccurs="1" name="PPTSlideShowState"
type="tns:PPTSlideShowState"/>
      <xs:element minOccurs="1" maxOccurs="1" name="PPTMediaPlaybackState"
type="tns:PPTMediaPlaybackState"/>
    </xs:sequence>
  </xs:complexType>
  <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:simpleType name="BroadcastAppCapability">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SessionTimeout"/>
      <xs:enumeration value="SessionIdleTimeOut"/>
      <xs:enumeration value="SupportVideo"/>
      <xs:enumeration value="SupportAudio"/>
      <xs:enumeration value="SupportNotes"/>
      <xs:enumeration value="MediaExtensions"/>
      <xs:enumeration value="MaxMediaSize"/>
    </xs:restriction>
  </xs:simpleType>

```

```

</xs:simpleType>
<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="BroadcastEndSession">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="user" type="tns:BroadcastUser"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="BroadcastUser">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="SessionId" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="UserToken" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="BroadcastEndSessionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastEndSessionResult"
type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BroadcastPutData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="user" type="tns:BroadcastUser"/>
      <xs:element minOccurs="0" maxOccurs="1" name="data"
type="tns:BroadcastDictionaryType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BroadcastPutDataResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastPutDataResult"
type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BroadcastGetAppCapabilities">
  <xs:complexType/>
</xs:element>
<xs:element name="BroadcastGetAppCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BroadcastGetAppCapabilitiesResult"
type="tns:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BroadcastPing">
  <xs:complexType/>

```

```

    </xs:element>
    <xs:element name="BroadcastPingResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" name="BroadcastPingResult"
type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="BroadcastDictionaryType">
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="item" type="tns:ItemType"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ItemType">
      <xs:sequence>
        <xs:element name="key" type="xs:string"/>
        <xs:element name="value" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="dictionary" type="tns:BroadcastDictionaryType"/>
  </xs:schema>
</wsdl:types>
<wsdl:message name="BroadcastStartSessionSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastStartSession"/>
</wsdl:message>
<wsdl:message name="BroadcastStartSessionSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastStartSessionResponse"/>
</wsdl:message>
<wsdl:message name="BroadcastEndSessionSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastEndSession"/>
</wsdl:message>
<wsdl:message name="BroadcastEndSessionSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastEndSessionResponse"/>
</wsdl:message>
<wsdl:message name="BroadcastPutDataSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastPutData"/>
</wsdl:message>
<wsdl:message name="BroadcastPutDataSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastPutDataResponse"/>
</wsdl:message>
<wsdl:message name="BroadcastGetAppCapabilitiesSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastGetAppCapabilities"/>
</wsdl:message>
<wsdl:message name="BroadcastGetAppCapabilitiesSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastGetAppCapabilitiesResponse"/>
</wsdl:message>
<wsdl:message name="BroadcastPingSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastPing"/>
</wsdl:message>
<wsdl:message name="BroadcastPingSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastPingResponse"/>
</wsdl:message>
<wsdl:portType name="PresentService_2_0Soap">
  <wsdl:operation name="BroadcastStartSession">
    <wsdl:input message="tns:BroadcastStartSessionSoapIn"/>
    <wsdl:output message="tns:BroadcastStartSessionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="BroadcastEndSession">

```

```

        <wsdl:input message="tns:BroadcastEndSessionSoapIn"/>
        <wsdl:output message="tns:BroadcastEndSessionSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastPutData">
        <wsdl:input message="tns:BroadcastPutDataSoapIn"/>
        <wsdl:output message="tns:BroadcastPutDataSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastGetAppCapabilities">
        <wsdl:input message="tns:BroadcastGetAppCapabilitiesSoapIn"/>
        <wsdl:output message="tns:BroadcastGetAppCapabilitiesSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="BroadcastPing">
        <wsdl:input message="tns:BroadcastPingSoapIn"/>
        <wsdl:output message="tns:BroadcastPingSoapOut"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PresentService_2_0Soap" type="tns:PresentService_2_0Soap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="BroadcastStartSession">
        <soap:operation
soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastStartSession"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="BroadcastEndSession">
        <soap:operation
soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastEndSession"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="BroadcastPutData">
        <soap:operation
soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPutData"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="BroadcastGetAppCapabilities">
        <soap:operation
soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastGetAppCapabiliti
es" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BroadcastPing">
    <soap:operation
      soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPing"
      style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="PresentService_2_0Soap12" type="tns:PresentService_2_0Soap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="BroadcastStartSession">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastStartSession"
      style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BroadcastEndSession">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastEndSession"
      style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BroadcastPutData">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPutData"
      style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BroadcastGetAppCapabilities">
    <soap12:operation
      soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastGetAppCapabilities"
      style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

```



```
<wsdl:operation name="BroadcastPing">
  <soap12:operation
soapAction="http://schemas.microsoft.com/server/broadcast/2010/main/BroadcastPing"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

Preliminary

7 Appendix B: Product Behavior

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

- Microsoft® Lync® Server 2013 Preview

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

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

8 Change Tracking

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

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

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

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

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

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

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

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

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

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

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

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

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

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.1.4.3.2.1 BroadcastPutData	Clarified the value descriptions for the SequenceNumber, FileVersion, and AppSpecificStateData keys.	N	Content updated.

9 Index

A

Abstract data model
[server](#) 22
[Applicability](#) 8
[AppServerInfo complex type](#) 10
[ArrayOfPPTBroadcastAnimationStepData complex type](#) 11
[ArrayOfPPTBroadcastMediaStateData complex type](#) 11
[Attribute groups](#) 20
[Attributes](#) 20

B

[BroadcastAppCapability simple type](#) 16
[BroadcastDataKey simple type](#) 16
[BroadcastState simple type](#) 17
[BroadcastUser complex type](#) 11

C

[Capability negotiation](#) 8
[Change tracking](#) 51
[ClientActions simple type](#) 18
[Complex types](#) 10
[AppServerInfo](#) 10
[ArrayOfPPTBroadcastAnimationStepData](#) 11
[ArrayOfPPTBroadcastMediaStateData](#) 11
[BroadcastUser](#) 11
[EnumTypes](#) 12
[PPTBroadcastAnimationStepData](#) 12
[PPTBroadcastMediaStateData](#) 13
[PPTStateData](#) 13
[ServiceError](#) 14
[ServiceResult](#) 14
[WordStateData](#) 15

D

Data model - abstract
[server](#) 22

E

[EnumTypes complex type](#) 12
Events
[local - server](#) 36
[timer - server](#) 35
Examples
[overview](#) 37
[Presenter client](#) 37

F

[Fields - vendor-extensible](#) 8
[Full WSDL](#) 42

G

[Glossary](#) 6
[Groups](#) 20

I

[Implementer - security considerations](#) 41
[Index of security parameters](#) 41
[Informative references](#) 7
Initialization
[server](#) 23
[Introduction](#) 6

L

Local events
[server](#) 36

M

Message processing
[server](#) 23
Messages
[AppServerInfo complex type](#) 10
[ArrayOfPPTBroadcastAnimationStepData complex type](#) 11
[ArrayOfPPTBroadcastMediaStateData complex type](#) 11
[attribute groups](#) 20
[attributes](#) 20
[BroadcastAppCapability simple type](#) 16
[BroadcastDataKey simple type](#) 16
[BroadcastState simple type](#) 17
[BroadcastUser complex type](#) 11
[ClientActions simple type](#) 18
[complex types](#) 10
[elements](#) 10
[enumerated](#) 9
[EnumTypes complex type](#) 12
[groups](#) 20
[namespaces](#) 9
[PPTBroadcastAnimationStepData complex type](#) 12
[PPTBroadcastMediaStateData complex type](#) 13
[PPTMediaPlaybackState simple type](#) 18
[PPTSlideShowState simple type](#) 19
[PPTStateData complex type](#) 13
[ServiceError complex type](#) 14
[ServiceErrorType simple type](#) 19
[ServiceResult complex type](#) 14
[simple types](#) 15
[syntax](#) 9
[transport](#) 9
[WordStateData complex type](#) 15

N

[Namespaces](#) 9
[Normative references](#) 6

O

Operations

[BroadcastEndSession](#) 23
[BroadcastGetAppCapabilities](#) 25
[BroadcastPing](#) 34
[BroadcastPutData](#) 28
[BroadcastStartSession](#) 31
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 41
[PPTBroadcastAnimationStepData complex type](#) 12
[PPTBroadcastMediaStateData complex type](#) 13
[PPTMediaPlaybackState simple type](#) 18
[PPTSlideShowState simple type](#) 19
[PPTStateData complex type](#) 13
[Preconditions](#) 8
[Prerequisites](#) 8
[Presenter client example](#) 37
[Product behavior](#) 50

R

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

S

Security

[implementer considerations](#) 41
[parameter index](#) 41

Sequencing rules

[server](#) 23

Server

[abstract data model](#) 22
[BroadcastEndSession operation](#) 23
[BroadcastGetAppCapabilities operation](#) 25
[BroadcastPing operation](#) 34
[BroadcastPutData operation](#) 28
[BroadcastStartSession operation](#) 31
[initialization](#) 23
[local events](#) 36
[message processing](#) 23
[sequencing rules](#) 23
[timer events](#) 35
[timers](#) 23
[ServiceError complex type](#) 14
[ServiceErrorType simple type](#) 19
[ServiceResult complex type](#) 14
[Simple types](#) 15
 [BroadcastAppCapability](#) 16
 [BroadcastDataKey](#) 16
 [BroadcastState](#) 17
 [ClientActions](#) 18
 [PPTMediaPlaybackState](#) 18

[PPTSlideShowState](#) 19
[ServiceErrorType](#) 19
[Standards assignments](#) 8
Syntax
 [messages - overview](#) 9

T

Timer events
 [server](#) 35
Timers
 [server](#) 23
[Tracking changes](#) 51
[Transport](#) 9
Types
 [complex](#) 10
 [simple](#) 15

V

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

W

[WordStateData complex type](#) 15
[WSDL](#) 42