

# [MS-PWBDPS]: PowerPoint Web Broadcast Discovery Protocol

---

## Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

## Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Minor	Clarified the meaning of the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.05	Minor	Clarified the meaning of the technical content.
11/15/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
04/11/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2013	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/30/2013	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.05	No change	No changes to the meaning, language, or formatting of the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
02/10/2014	1.05	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Protocol Overview (Synopsis)	7
1.4 Relationship to Other Protocols	7
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
<b>2 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	9
2.2.4 Complex Types	10
2.2.4.1 BroadcastHostInfo	10
2.2.4.2 Version	10
2.2.5 Simple Types	11
2.2.6 Attributes	11
2.2.7 Groups	11
2.2.8 Attribute Groups	11
<b>3 Protocol Details</b>	<b>12</b>
3.1 Server Details	12
3.1.1 Abstract Data Model	12
3.1.2 Timers	12
3.1.3 Initialization	12
3.1.4 Message Processing Events and Sequencing Rules	13
3.1.4.1 BroadcastGetHostInfo	13
3.1.4.1.1 Messages	13
3.1.4.1.1.1 BroadcastGetHostInfoSoapIn	13
3.1.4.1.1.2 BroadcastGetHostInfoSoapOut	13
3.1.4.1.2 Elements	13
3.1.4.1.2.1 BroadcastGetHostInfo	14
3.1.4.1.2.2 BroadcastGetHostInfoResponse	14
3.1.4.1.3 Complex Types	14
3.1.4.1.3.1 BroadcastProtocolInfo	14
3.1.4.1.3.2 ClientInfo	15
3.1.4.1.3.3 ArrayOfVersion	15
3.1.4.1.3.4 ServiceResult	15
3.1.4.1.3.5 ServiceError	16
3.1.4.1.4 Simple Types	16
3.1.4.1.4.1 ServiceErrorType	16
3.1.4.1.4.2 ClientActions	17
3.1.4.1.5 Attributes	17

3.1.4.1.6	Groups.....	17
3.1.4.1.7	Attribute Groups .....	17
3.1.5	Timer Events .....	18
3.1.6	Other Local Events .....	18
<b>4</b>	<b>Protocol Examples.....</b>	<b>19</b>
4.1	Sending Protocol Client Information to the Protocol Server.....	19
<b>5</b>	<b>Security.....</b>	<b>21</b>
5.1	Security Considerations for Implementers.....	21
5.2	Index of Security Parameters .....	21
<b>6</b>	<b>Appendix A: Full WSDL.....</b>	<b>22</b>
<b>7</b>	<b>Appendix B: Product Behavior.....</b>	<b>25</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>26</b>
<b>9</b>	<b>Index .....</b>	<b>27</b>

# 1 Introduction

This document specifies the PowerPoint Web Broadcast Discovery Protocol, which enables a protocol client to obtain information from a protocol server about an appropriate endpoint to use to access a PowerPoint Web Broadcast Host 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)**  
**SOAP**  
**SOAP action**  
**SOAP body**  
**SOAP fault**  
**XML namespace**

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

**Uniform Resource Locator (URL)**  
**Web Services Description Language (WSDL)**  
**website**  
**WSDL message**  
**WSDL operation**  
**XML namespace prefix**

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 documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-PWBHPS] Microsoft Corporation, "[PowerPoint Web Broadcast Host Protocol](#)".

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

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

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

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

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

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

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

[XMLSCHEMA1] Thompson, H.S., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

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

## 1.2.2 Informative References

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

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

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

[RFC2822] Resnick, P., Ed., "Internet Message Format", STD 11, RFC 2822, April 2001, <http://www.ietf.org/rfc/rfc2822.txt>

## 1.3 Protocol Overview (Synopsis)

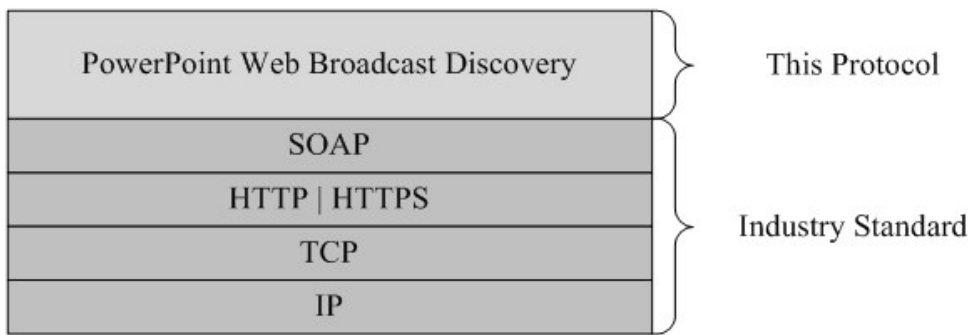
This protocol enables a protocol client to send a request containing client information to the protocol server and then receive information about the endpoint that is most suited for future communication with the protocol client.

The protocol server uses the information provided by the protocol client such as the list of supported protocol versions to decide a suitable endpoint.

## 1.4 Relationship to Other Protocols

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

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



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

## 1.5 Prerequisites/Preconditions

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

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

## 1.6 Applicability Statement

This protocol is designed to retrieve information about an appropriate endpoint to use for a given client.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP as specified in section [2.1](#).

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.



## 2 Messages

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

### 2.1 Transport

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

Protocol messages **MUST** be formatted as specified either in [\[SOAP1.1\]](#) section 4 or in [\[SOAP1.2/1\]](#), section 5. Protocol server faults **MUST** be returned either using HTTP Status Codes, as specified in [\[RFC2616\]](#) section 10 or using **SOAP faults**, as specified in either [\[SOAP1.1\]](#) section 4.4 or in [\[SOAP1.2/1\]](#) section 5.4.

### 2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses XML Schema as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and 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
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a> <a href="#">[XMLSCHEMA2]</a>
soap12	<a href="http://schemas.xmlsoap.org/wsdl/soap12/">http://schemas.xmlsoap.org/wsdl/soap12/</a>	<a href="#">[SOAP1.2/1]</a> <a href="#">[SOAP1.2/2]</a>
tns	<a href="http://schemas.microsoft.com/server/powerpoint/2009/main">http://schemas.microsoft.com/server/powerpoint/2009/main</a>	
soap	<a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a>	<a href="#">[SOAP1.1]</a>

#### 2.2.2 Messages

None.

#### 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
BroadcastHostInfo	This complex type specifies the PowerPoint Web Broadcast Host Protocol endpoint information returned by the protocol server to the protocol client, as specified in <a href="#">[MS-PWBHPS]</a> .
Version	This complex type specifies the protocol version information.

### 2.2.4.1 BroadcastHostInfo

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

This complex type specifies the PowerPoint Web Broadcast Host Protocol endpoint information returned by the protocol server to the protocol client, as specified in [\[MS-PWBHPS\]](#).

```
<xs:complexType name="BroadcastHostInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ProtocolVersion" type="tns:Version"/>
    <xs:element minOccurs="0" maxOccurs="1" name="HostUrl" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

**ProtocolVersion:** A [Version](#) element that specifies the version number of the protocol used by the server that is referenced by the **HostUrl** element. This element **MUST** be present.

**HostUrl:** A string element, as specified in [\[XMLSCHEMA2\]](#) section 3.2.1, that specifies a URL of a PowerPoint Web Broadcast Host Protocol server, as specified in [\[MS-PWBHPS\]](#). This element **MUST** be present.

### 2.2.4.2 Version

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

This complex type specifies the protocol version information.

```
<xs:complexType name="Version">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="MajorNumber" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="MinorNumber" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

**MajorNumber:** An int element, as specified in [\[XMLSCHEMA2\]](#) section 3.3.17, that specifies the major number of the protocol version. This element **MUST** be present.

**MinorNumber:** An int element, as specified in [\[XMLSCHEMA2\]](#) section 3.3.17, that specifies the minor number of the protocol version. This element **MUST** be present.

### **2.2.5 Simple Types**

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

### **2.2.6 Attributes**

None.

### **2.2.7 Groups**

None.

### **2.2.8 Attribute Groups**

None.

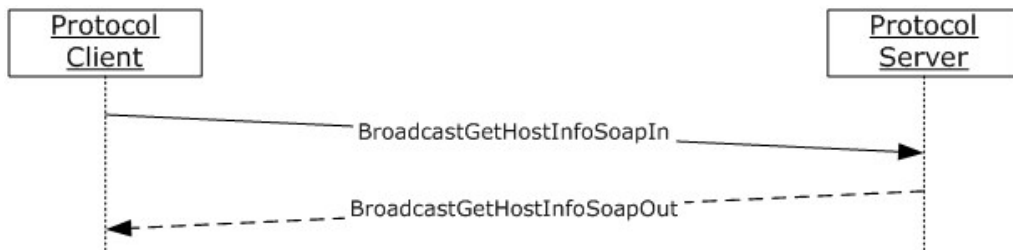
### 3 Protocol Details

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

#### 3.1 Server Details

The following diagram describes the communication between the protocol client and the protocol server:



**Figure 2: Communication between protocol client and protocol server**

The protocol client sends a [BroadcastGetHostInfoSoapIn](#) SOAP message to the protocol server. In the message the protocol client sends information about the client name, client version and the list of PowerPoint Web Broadcast Host Protocol versions supported by the protocol client, as specified in [\[MS-PWBHPS\]](#).

The protocol server responds by sending a [BroadcastGetHostInfoSoapOut](#) SOAP message to the protocol client. The message contains information about the most suitable endpoint for the protocol client to use to access a PowerPoint Web Broadcast Host Protocol server, as specified in [\[MS-PWBHPS\]](#). The message also contains the PowerPoint Web Broadcast Host Protocol version supported by the endpoint, as specified in [\[MS-PWBHPS\]](#).

##### 3.1.1 Abstract Data Model

None.

##### 3.1.2 Timers

None.

##### 3.1.3 Initialization

The protocol server **MUST** expose its Web methods at a URL that is known by the protocol client.

### 3.1.4 Message Processing Events and Sequencing Rules

For specifications about the sequencing of protocol messages and how they relate to each other, see section [3.1](#). The following sections specify the details of each individual message.

The protocol server SHOULD return the most suitable PowerPoint Web Broadcast Host endpoint for the protocol client to use, depending on the protocol client's information (name and version) and the PowerPoint Web Broadcast Host Protocol versions supported by the protocol client, as specified in [\[MS-PWBHPS\]](#). The method MUST NOT throw an exception.

This specification includes the following **WSDL operations**:

WSDL Operation	Description
BroadcastGetHostInfo	This web method is used by the protocol client to retrieve host information from the protocol server.

#### 3.1.4.1 BroadcastGetHostInfo

This web method is used by the protocol client to retrieve host information from the protocol server.

```
<wsdl:operation name="BroadcastGetHostInfo">
  <wsdl:input message="tns:BroadcastGetHostInfoSoapIn"/>
  <wsdl:output message="tns:BroadcastGetHostInfoSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **BroadcastGetHostInfoSoapIn** request message, and the protocol server responds with a **BroadcastGetHostInfoSoapOut** response message as specified in the following sections.

##### 3.1.4.1.1 Messages

The following **WSDL message** definitions are specific to this operation.

###### 3.1.4.1.1.1 BroadcastGetHostInfoSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/server/powerpoint/2009/main/BroadcastGetHostInfo
```

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

###### 3.1.4.1.1.2 BroadcastGetHostInfoSoapOut

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

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

##### 3.1.4.1.2 Elements

The following XML Schema element definitions are specific to this operation.

### 3.1.4.1.2.1 BroadcastGetHostInfo

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

```
<xs:element name="BroadcastGetHostInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="broadcastProtocolInfo"
type="tns:BroadcastProtocolInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**broadcastProtocolInfo:** A [BroadcastProtocolInfo](#) element that specifies the protocol client identity information and the list of PowerPoint Web Broadcast Host Protocol versions supported by the protocol client, as specified in [\[MS-PWBHPS\]](#). This element **MUST** be present.

### 3.1.4.1.2.2 BroadcastGetHostInfoResponse

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

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

**BroadcastGetHostInfoResult:** A [ServiceResult](#) element that specifies the result of the operation. This element **MUST** be present. The **Result** child element **MUST** be a [BroadcastHostInfo](#) if the [BroadcastGetHostInfo](#) operation succeeds.

### 3.1.4.1.3 Complex Types

The following XML Schema complex type definitions are specific to this operation.

#### 3.1.4.1.3.1 BroadcastProtocolInfo

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

This complex type specifies the protocol client identity information and the list of protocol versions supported by the protocol client.

```
<xs:complexType name="BroadcastProtocolInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ClientInfo" type="tns:ClientInfo"/>
    <xs:element minOccurs="0" maxOccurs="1" name="VersionList" type="tns:ArrayOfVersion"/>
  </xs:sequence>
</xs:complexType>
```

**ClientInfo:** A [ClientInfo](#) element that specifies the name and version of the protocol client. This element **MUST** be present.

**VersionList:** An [ArrayOfVersion](#) element that specifies the list of PowerPoint Web Broadcast Host Protocol versions supported by the protocol client, as specified in [\[MS-PWBHPS\]](#). This element **MUST** be present.

### 3.1.4.1.3.2 ClientInfo

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

This complex type specifies the protocol client's name and version.

```
<xs:complexType name="ClientInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Version" type="tns:Version"/>
  </xs:sequence>
</xs:complexType>
```

**Name:** A string element, as specified in [\[XMLSCHEMA2\]](#) section 3.2.1, that specifies the name of the protocol client. This element **MUST** be present.

**Version:** A [Version](#) element that specifies the version of the protocol client. This element **MUST** be present.

### 3.1.4.1.3.3 ArrayOfVersion

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

This complex type specifies a list of [Version](#) elements.

```
<xs:complexType name="ArrayOfVersion">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Version" nillable="true"
type="tns:Version"/>
  </xs:sequence>
</xs:complexType>
```

**Version:** Each element **MUST** specify a Version element.

### 3.1.4.1.3.4 ServiceResult

**Namespace:** <http://schemas.microsoft.com/server/powerpoint/2009/main>

The complex type 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">
  <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 **anyType** element, as specified in [\[XMLSCHEMA1\]](#) section 3.4.7, that specifies a successful result of a protocol message response. The **Result** child element **MUST NOT** be empty if

the BroadcastHostInfo operation succeeds and if the **Error** element is empty. Otherwise, this element MUST be empty.

**Error:** A [ServiceError](#) element that specifies an error result of a protocol message response. This element MUST NOT be empty if the **Result** element is empty. Otherwise, this element MUST be empty.

### 3.1.4.1.3.5 ServiceError

**Namespace:** http://schemas.microsoft.com/server/powerpoint/2009/main

This complex type specifies error information returned by the protocol server to a protocol client.

```
<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>
```

**Message:** A string element, as specified in [XMLSCHEMA2](#) section 3.2.1, that specifies the error message description. The string length MUST be greater than zero if the **Type** element has a value of [ApplicationError](#). The string length MAY be equal to zero if the **Type** element has a value of UnknownError or Timeout. This element MUST be present.

**Title:** A string element, as specified in [XMLSCHEMA2](#) section 3.2.1, that specifies the error title. The string length MUST be greater than zero if the **Type** element has a value of ApplicationError. The string length MAY be equal to zero if the **Type** element has a value of UnknownError or Timeout. This element MUST be present.

**Type:** A ServiceErrorType element that specifies the error type. This element MUST be present.

**RecommendedActions:** Reserved. MUST be ignored.

### 3.1.4.1.4 Simple Types

The following XML Schema simple type definitions are specific to this operation.

#### 3.1.4.1.4.1 ServiceErrorType

**Namespace:** http://schemas.microsoft.com/server/powerpoint/2009/main

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

```
<xs:simpleType name="ServiceErrorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UnknownError"/>
    <xs:enumeration value="ApplicationError"/>
    <xs:enumeration value="Timeout"/>
  </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 time-out.

### 3.1.4.1.4.2 ClientActions

**Namespace:** http://schemas.microsoft.com/server/powerpoint/2009/main

Reserved. MUST be ignored.

```
<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>
```

The following table specifies the allowable values for ClientActions:

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

### 3.1.4.1.5 Attributes

None.

### 3.1.4.1.6 Groups

None.

### 3.1.4.1.7 Attribute Groups

None.

### **3.1.5 Timer Events**

None.

### **3.1.6 Other Local Events**

None.

## 4 Protocol Examples

The following example contains a sample interaction between the protocol client and the protocol server.

### 4.1 Sending Protocol Client Information to the Protocol Server

The following example demonstrates the interaction between the protocol client and the protocol server. For the sake of succinctness, only the SOAP body is listed.

In this example, the protocol client sends the following information to the protocol server, as specified in [\[MS-PWBHPS\]](#):

- Name and version of the protocol client.
- The PowerPoint Web Broadcast Host Protocol version understood by the protocol client,.

The following **BroadcastGetHostInfoSoapIn** message is sent to the protocol server:

```
<BroadcastGetHostInfo xmlns="http://schemas.microsoft.com/server/powerpoint/2009/main">
  <broadcastProtocolInfo>
    <ClientInfo>
      <Name>Microsoft PowerPoint</Name>
      <Version>
        <MajorNumber>14</MajorNumber>
        <MinorNumber>0</MinorNumber>
      </Version>
    </ClientInfo>
    <VersionList>
      <Version>
        <MajorNumber>1</MajorNumber>
        <MinorNumber>0</MinorNumber>
      </Version>
    </VersionList>
  </broadcastProtocolInfo>
</BroadcastGetHostInfo>
```

The protocol server looks at the information passed by the protocol client and sends an appropriate PowerPoint Web Broadcast Host Protocol endpoint and its version information to the protocol client, as specified in [\[MS-PWBHPS\]](#). The following **BroadcastGetHostInfoSoapOut** message is sent to the protocol client:

```
<BroadcastGetHostInfoResponse
xmlns="http://schemas.microsoft.com/server/powerpoint/2009/main">
  <BroadcastGetHostInfoResult>
    <Result xsi:type="BroadcastHostInfo">
      <ProtocolVersion>
        <MajorNumber>1</MajorNumber>
        <MinorNumber>0</MinorNumber>
      </ProtocolVersion>
      <HostUrl>http://www.contoso.com/PowerPointBroadcastHost_1_0.asmx?wsdl</HostUrl>
    </Result>
  </BroadcastGetHostInfoResult>
</BroadcastGetHostInfoResponse>
```

The protocol client then talks to the endpoint passed by the protocol server if the protocol client is compatible with the protocol version of the endpoint.

## 5 Security

### 5.1 Security Considerations for Implementers

There are no security considerations that are specific to this protocol. General security considerations pertaining to [RFC2822](#) apply.

This protocol does not introduce any additional security considerations beyond those that apply to its underlying protocols.

### 5.2 Index of Security Parameters

None.

## 6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/server/powerpoint/2009/main"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://schemas.microsoft.com/server/powerpoint/2009/main"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/server/powerpoint/2009/main">
      <s:element name="BroadcastGetHostInfo">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="broadcastProtocolInfo"
type="tns:BroadcastProtocolInfo" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="BroadcastProtocolInfo">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="ClientInfo" type="tns:ClientInfo" />
          <s:element minOccurs="0" maxOccurs="1" name="VersionList" type="tns:ArrayOfVersion"
/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="ClientInfo">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="Name" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="Version" type="tns:Version" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Version">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="MajorNumber" type="s:int" />
          <s:element minOccurs="1" maxOccurs="1" name="MinorNumber" type="s:int" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="ArrayOfVersion">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Version" nillable="true"
type="tns:Version" />
        </s:sequence>
      </s:complexType>
      <s:element name="BroadcastGetHostInfoResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="BroadcastGetHostInfoResult"
type="tns:ServiceResult" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ServiceResult">
```

```

    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="Result" />
      <s:element minOccurs="0" maxOccurs="1" name="Error" type="tns:ServiceError" />
    </s:sequence>
  </s:complexType>
  <s:complexType name="ServiceError">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="Message" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
      <s:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ServiceErrorType" />
      <s:element minOccurs="1" maxOccurs="1" name="RecommendedActions"
type="tns:ClientActions" />
    </s:sequence>
  </s:complexType>
  <s:simpleType name="ServiceErrorType">
    <s:restriction base="s:string">
      <s:enumeration value="UnknownError" />
      <s:enumeration value="ApplicationError" />
      <s:enumeration value="Timeout" />
    </s:restriction>
  </s:simpleType>
  <s:simpleType name="ClientActions">
    <s:list>
      <s:simpleType>
        <s:restriction base="s:string">
          <s:enumeration value="None" />
          <s:enumeration value="Dismiss" />
          <s:enumeration value="Close" />
          <s:enumeration value="OpenInClient" />
          <s:enumeration value="Refresh" />
        </s:restriction>
      </s:simpleType>
    </s:list>
  </s:simpleType>
  <s:complexType name="BroadcastHostInfo">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="ProtocolVersion" type="tns:Version" />
      <s:element minOccurs="0" maxOccurs="1" name="HostUrl" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="BroadcastGetHostInfoSoapIn">
  <wsdl:part name="parameters" element="tns:BroadcastGetHostInfo" />
</wsdl:message>
<wsdl:message name="BroadcastGetHostInfoSoapOut">
  <wsdl:part name="parameters" element="tns:BroadcastGetHostInfoResponse" />
</wsdl:message>
<wsdl:portType name="PowerpointBroadcastHostWebServiceSoap">
  <wsdl:operation name="BroadcastGetHostInfo">
    <wsdl:input message="tns:BroadcastGetHostInfoSoapIn" />
    <wsdl:output message="tns:BroadcastGetHostInfoSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PowerpointBroadcastHostWebServiceSoap"
type="tns:PowerpointBroadcastHostWebServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="BroadcastGetHostInfo">

```

```
<soap:operation
soapAction="http://schemas.microsoft.com/server/powerpoint/2009/main/BroadcastGetHostInfo"
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="PowerpointBroadcastHostWebServiceSoap12"
type="tns:PowerpointBroadcastHostWebServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="BroadcastGetHostInfo">
    <soap12:operation
soapAction="http://schemas.microsoft.com/server/powerpoint/2009/main/BroadcastGetHostInfo"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```



## 7 Appendix B: Product Behavior

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

- Microsoft PowerPoint 2010
- Microsoft PowerPoint Web App

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

No table of changes is available. The document is either new or has had no changes since its last release.

## 9 Index

### A

Abstract data model  
    [server](#) 12  
[Applicability](#) 8  
[Attribute groups](#) 11  
[Attributes](#) 11

### B

[BroadcastHostInfoComplex type](#) 10

### C

[Capability negotiation](#) 8  
[Change tracking](#) 26  
[Complex types](#) 10  
    [BroadcastHostInfo](#) 10  
    [Version](#) 10

### D

Data model - abstract  
    [server](#) 12

### E

Events  
    [local - server](#) 18  
    [timer - server](#) 18  
Examples  
    [overview](#) 19  
    [sending protocol client information to the protocol server](#) 19

### F

[Fields - vendor-extensible](#) 8  
[Full WSDL](#) 22

### G

[Glossary](#) 6  
[Groups](#) 11

### I

[Implementer - security considerations](#) 21  
[Index of security parameters](#) 21  
[Informative references](#) 7  
Initialization  
    [server](#) 12  
[Introduction](#) 6

### L

Local events  
    [server](#) 18

### M

Message processing  
    [server](#) 13  
Messages  
    [attribute groups](#) 11  
    [attributes](#) 11  
    [BroadcastHostInfoComplex type](#) 10  
    [complex types](#) 10  
    [elements](#) 9  
    [enumerated](#) 9  
    [groups](#) 11  
    [namespaces](#) 9  
    [simple types](#) 11  
    [syntax](#) 9  
    [transport](#) 9  
    [VersionComplex type](#) 10

### N

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

### O

Operations  
    [BroadcastGetHostInfo](#) 13  
[Overview \(synopsis\)](#) 7

### P

[Parameters - security index](#) 21  
[Preconditions](#) 8  
[Prerequisites](#) 8  
[Product behavior](#) 25  
Protocol Details  
    [overview](#) 12

### R

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

### S

Security  
    [implementer considerations](#) 21  
    [parameter index](#) 21  
    [Sending protocol client information to the protocol server example](#) 19  
Sequencing rules  
    [server](#) 13  
Server  
    [abstract data model](#) 12  
    [BroadcastGetHostInfo operation](#) 13  
    [details](#) 12

- [initialization](#) 12
- [local events](#) 18
- [message processing](#) 13
- [sequencing rules](#) 13
- [timer events](#) 18
- [timers](#) 12
- [Simple types](#) 11
- [Standards assignments](#) 8
- Syntax
  - [messages - overview](#) 9

## T

- Timer events
  - [server](#) 18
- Timers
  - [server](#) 12
- [Tracking changes](#) 26
- [Transport](#) 9
- Types
  - [complex](#) 10
  - [simple](#) 11

## V

- [Vendor-extensible fields](#) 8
- [Versioncomplex type](#) 10
- [Versioning](#) 8

## W

- [WSDL](#) 22