

[MS-OMPWHP]: Office Mobile PowerPoint Web Handler Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
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	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.06	Editorial	Changed language and formatting in the technical content.
12/17/2010	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.06	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	Major	Significantly changed the technical content.
04/11/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	6
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Common Message Syntax	8
2.2.1 Namespaces	8
2.2.2 Messages	8
2.2.3 Elements	8
2.2.4 Complex Types	8
2.2.5 Simple Types	8
2.2.6 Attributes	8
2.2.7 Groups	8
2.2.8 Attribute Groups	8
3 Protocol Details	9
3.1 Server Details	9
3.1.1 Abstract Data Model	9
3.1.2 Timers	9
3.1.3 Initialization	10
3.1.4 Message Processing Events and Sequencing Rules	10
3.1.4.1 MobilePresentation	10
3.1.4.1.1 GetMetadataXml	10
3.1.4.1.1.1 Return Values	11
3.1.4.1.2 GetLatestDocumentVersion	11
3.1.4.1.2.1 Return Values	11
3.1.4.2 MobileSlideInfo	12
3.1.4.2.1 GetSlideInfoXml	12
3.1.4.2.1.1 Return Values	12
3.1.4.3 MobileSlideImage	12
3.1.4.3.1 GetSlideImage	12
3.1.4.3.1.1 Return Values	13
3.1.5 Timer Events	13
3.1.6 Other Local Events	13
4 Protocol Examples	14
4.1 Presentation Information	14
4.2 Slide Image	15
4.3 Slide Information	15
5 Security	17

5.1 Security Considerations for Implementers	17
5.2 Index of Security Parameters	17
6 Appendix A: Full WSDL	18
7 Appendix B: Product Behavior	19
8 Change Tracking.....	20
9 Index	22

1 Introduction

This document specifies the Office Mobile PowerPoint Web Handler Protocol, which enables a protocol client to obtain information about presentation content 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)
XML

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

HTTP GET
presentation
presentation broadcast
presentation slide
Uniform Resource Locator (URL)
XML namespace
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-PWBHPS] Microsoft Corporation, "[PowerPoint Web Broadcast Host Protocol Specification](#)".

[MS-PWVRSC] Microsoft Corporation, "[PowerPoint Web Viewer Rendered Static Content Structure Specification](#)".

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

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

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

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

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

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

1.2.2 Informative References

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to send a request to retrieve **presentation** content from the protocol server.

To facilitate this, the protocol allows the protocol client to request specific pieces of content from a presentation stored on the protocol server. In a **presentation slide** contained within a presentation, a protocol client can retrieve information describing the presentation slide contents as well as images of the presentation slide contents.

1.4 Relationship to Other Protocols

This protocol uses **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:

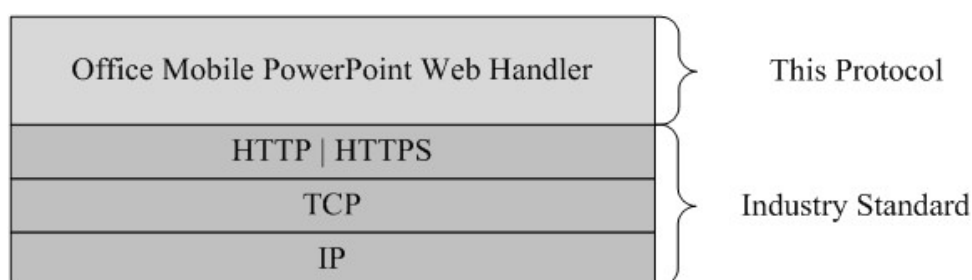


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a site that is identified by a **URL** that is known by protocol clients.

The protocol server endpoint to send requests for the presentation metadata is formed by appending "/_layouts/MobilePresentation.ashx" to the URL of the site, for example:
http://www.contoso.com/Repository/_layouts/MobilePresentation.ashx.

The protocol server endpoint to send requests for the presentation slide information is formed by appending the value of the **url** attribute of the **CT_SlideInfo** element ([\[MS-PWVRSC\]](#) section 2.3.11) contained in the metadata for the requested presentation to the URL of the site, for example: http://www.contoso.com/Repository/_layouts/MobileSlideInfo.ashx.

The protocol server endpoint to send requests for the presentation slide image is formed by appending the value of the **url** attribute of the **CT_SlideImage** element ([\[MS-PWVRSC\]](#) section 2.3.10) contained in the metadata for the requested presentation to the URL of the site, for example: http://www.contoso.com/Repository/_layouts/MobileSlideImage.ashx.

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

1.6 Applicability Statement

This protocol is designed to retrieve information about presentations that are stored on the protocol server.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses HTTP(S) version 1.1 (as specified in [RFC2616](#)) as transport for the **HTTP GET** method.

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
xsd	http://www.w3.org/2001/XMLSchema	XMLSCHEMA1 XMLSCHEMA2

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

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

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

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults using HTTP status codes.

3.1 Server Details

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

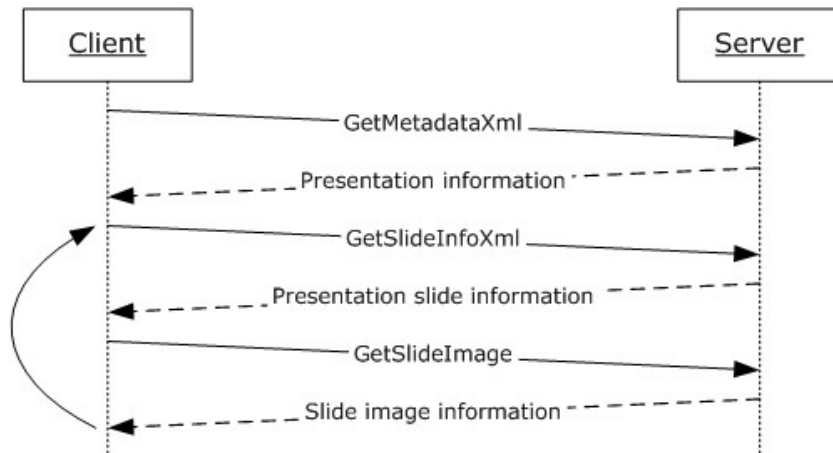


Figure 2: Sample communication between protocol client and protocol server

First, a protocol client calls **GetMetadataXml** and the protocol server responds with the information about the requested presentation.

The protocol client, using the preceding information, makes one or more calls to **GetSlideInfoXml**, to retrieve information about the requested presentation slide, and **GetSlideImage**, to retrieve the image of the requested presentation slide.

3.1.1 Abstract Data Model

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

presentationId: An entity which represents a unique identifier for a presentation.

3.1.2 Timers

None.

3.1.3 Initialization

The protocol server MUST expose its Web methods at a URL, which builds upon a base URL.

The URL MUST be constructed as base URL/_layouts/handler name.

The base URL can be any Web site URL, such as "http://www.contoso.com/Repository". The part of _layouts/handler name can be any of the following:

```
_layouts/MobilePresentation.ashx  
_layouts/MobileSlideImage.ashx  
_layouts/MobileSlideInfo.ashx
```

This is the minimal required structure. Case-sensitivity is specific to the protocol server implementation.

3.1.4 Message Processing Events and Sequencing Rules

The following table describes the supported operations.

Operation	Description
GetMetadataXml	Retrieves the metadata for a presentation.
GetLatestDocumentVersion	Retrieves the current version of a presentation.
GetSlideImage	Retrieves the image of a presentation slide.
GetSlideInfoXml	Retrieves information about content of a presentation slide.

3.1.4.1 MobilePresentation

The methods in this section are supported by MobilePresentation.ashx.

3.1.4.1.1 GetMetadataXml

This method retrieves the metadata for a presentation. In cases other than that of a **presentation broadcast**, the URL to this method is constructed by appending _layouts/MobilePresentation.ashx to the site name. The parameter to be passed while calling this method is:

presentationId: An **xsd:string** element ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the absolute or relative URL to the presentation.

Also, the scheme of the URL MUST match the scheme of the **presentationId** if the **presentationId** is being passed as an absolute URL.

In case of a presentation broadcast the URL to this method is constructed by replacing PowerpointFrame.aspx with MobilePresentation.ashx in the attendee URL. An attendee URL is contained in the **BroadcastGetAttendeeUrlResponse** defined in [\[MS-PWBHPS\]](#) section 3.1.4.2.2.2. No additional parameters need to be passed to the protocol server in this case.

If any of the preceding conditions are not satisfied, the method returns an error.

3.1.4.1.1.1 Return Values

This method MUST send back to the protocol client an HTTP response that MUST be in the following table.

Value	Description
200	If successful, the protocol server returns an XML string specifying the presentation metadata, which contains a mobilePres root element as specified in [MS-PWVRSC] section 2.1.1. An X-DocVersion HTTP header is present (except in the case of a presentation broadcast) which specifies the version of the presentation. An X-DocUrl HTTP header is present (except in the case of a presentation broadcast) which specifies the URL of the presentation. All other cases MUST be treated as failure by the protocol client.
503	Failure. The server also sets an HTTP header X-Error xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) in the response. If $100 \leq \text{value} < 200$, the protocol client SHOULD retry this call, else this header MUST be ignored.

3.1.4.1.2 GetLatestDocumentVersion

This method retrieves the version of the presentation. In cases other than that of a presentation broadcast, the URL to this method is constructed by appending `_layouts/MobilePresentation.ashx` to the site name. The parameters to be passed while calling this method are as follows:

presentationId: An **xsd:string** element ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the URL to the presentation. This parameter is optional. If the parameter is specified, the URL MUST be absolute or relative. If the parameter is not specified or if it is empty, then it MUST be auto generated.

getCurrentDocVersion: An **xsd:string** element ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies if the version of the document is to be returned. If this parameter is set to **true**, the server MUST return the version. For any other legal string, the server MUST ignore this parameter and treat this call as the **GetMetaDataXml** method (section [3.1.4.1.1](#)) call.

Also, the scheme of the URL MUST match the scheme of the **presentationId** if the **presentationId** is being passed as an absolute URL.

In case of a presentation broadcast, the URL to this method is constructed by replacing `PowerpointFrame.aspx` with `MobilePresentation.ashx` in the attendee URL. An attendee URL is contained in the **BroadcastGetAttendeeUrlResponse** defined in [\[MS-PWBHPS\]](#) section 3.1.4.3.2.2. Only the **getCurrentDocVersion** parameter needs to be passed to the protocol server in this case.

If any of the preceding conditions about the parameters are not satisfied, the method returns an error.

3.1.4.1.2.1 Return Values

This method MUST send back to the protocol client an HTTP response that MUST be in the following table.

Value	Description
200	If successful, the protocol server sets an X-DocVersion HTTP header which is an xsd:string ([XMLSCHEMA2] section 3.2.1) and specifies the version of the presentation. An X-DocUrl HTTP header is present (except in the case of a presentation broadcast) which specifies the URL of the

Value	Description
	presentation. All other cases MUST be treated as failure by the protocol client.
503	Failure. The server also sets an HTTP header X-Error xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) in the response. If 100 <= value < 200, the protocol client SHOULD retry this call, else this header MUST be ignored.

3.1.4.2 MobileSlideInfo

The methods in this section are supported by MobileSlideInfo.ashx.

3.1.4.2.1 GetSlideInfoXml

This method retrieves the information for a particular presentation slide. The URL to this method is constructed by appending "_layouts/" and the value of the **url** attribute of the **slideInfo** element in the **mobilePres** root element returned by the **GetMetadataXml** method, as specified in [MS-PWVRSC] section 2.3.11, to the URL of the site. The parameters to be passed while calling this function are as follows:

infoId: An **xsd:string** element ([XMLSCHEMA2] section 3.2.1) that specifies the presentation slide for which information is to be retrieved. It MUST be set to the value of the **sldInfo** attribute of a **CT_SlideEntry** element specified in [MS-PWVRSC] section 2.3.1.

If any of the preceding conditions about the parameters are not satisfied, the method returns an error.

3.1.4.2.1.1 Return Values

This method MUST send back to the protocol client an HTTP response that MUST be in the following table.

Value	Description
200	If successful, the protocol server returns an XML string specifying information about the requested presentation slide, which contains a sld root element as specified in [MS-PWVRSC] section 2.1.3. All other cases MUST be treated as failure by the protocol client.
503	Failure. The server also sets an HTTP header X-Error xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) in the response. If 100 <= value < 200, the protocol client SHOULD retry this call, else this header MUST be ignored.

3.1.4.3 MobileSlideImage

The methods in this section are supported by MobileSlideImage.ashx.

3.1.4.3.1 GetSlideImage

This method retrieves the image for a particular presentation slide in a presentation. The URL to this method is constructed by appending "_layouts/" and the value of the **url** attribute of **slideImage** element in the **mobilePres** root element returned by the **GetMetadataXml** method, as specified in [MS-PWVRSC] section 2.3.10, to the URL of the site. The parameters to be passed while calling this function are as follows:

imageId: An **xsd:string** element ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the presentation slide for which information is to be retrieved. It MUST be set to the value of the **sldImg** attribute of a **CT_SlideEntry** element specified in [\[MS-PWVRSC\]](#) section 2.3.1.

width: An **xsd:unsignedInt** element ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the width of the requested image in pixels. The maximum permissible value of this parameter is 1300. If the specified value is 0, a value of 1 MUST be used.

height: An **xsd:unsignedInt** element ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the height of the requested image in pixels. The maximum permissible value of this parameter is 1300. If the specified value is 0, a value of 1 MUST be used.

format: An **xsd:string** element ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the image format of the requested image. The only acceptable values of this parameter are **png** and **jpeg**. This parameter is optional. The default value of this parameter is **jpeg**. This default value of **jpeg** will be used if any other values other than the acceptable values are used.

If any of the preceding conditions about the parameters are not satisfied, the method returns an error.

3.1.4.3.1.1 Return Values

This method MUST send back to the protocol client an HTTP response that MUST be in the following table.

Value	Description
200	If successful, the protocol server returns an image of the requested page in the specified format. All other cases MUST be treated as failure by the protocol client.
503	Failure. The server also sets an HTTP header X-Error xsd:unsignedInt ([XMLSCHEMA2] section 3.3.22) in the response. If $100 \leq \text{value} < 200$, the protocol client SHOULD retry this call, else this header MUST be ignored.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples contain a sample interaction between the protocol client and the protocol server. These examples are illustrative of the preceding specification and they do not cover all possible structure usage scenarios. They are not intended to replace the preceding specification but rather to clarify and enhance it. In the following examples, the client wants to view a presentation located at <http://sptestamd/Documents/Demo.pptx>.

4.1 Presentation Information

The URL for **GetMetadataXml** (section [3.1.4.1.1](#)) is constructed as http://sptestamd/_layouts/MobilePresentation.ashx?presentationId=http://sptestamd/Documents/Demo.pptx.

The protocol client then performs an HTTP GET on this URL.

The protocol server responds with a **CT_MobilePres** ([\[MS-PWVRSC\]](#) section 2.3.12) element which contains information about the presentation.

```
<?xml version="1.0" encoding="utf-8"?>
<p:mobilePres xmlns:p="http://schemas.microsoft.com/server/powerpoint/2009/mobile">
  <p:slideImage
    url="/MobileSlideImage.ashx?presentationId=d%3DH%253Asptestamd%252F%252FDocuments%252FDemo%252E
    pptx%26z%3D79AF7428%2D11BC%2D4D49%2DA0D8%2D4AC7DBA09A951" image="imageId" width="width"
    height="height" format="format" />
  <p:slideInfo
    url="/MobileSlideInfo.ashx?presentationId=d%3DH%253Asptestamd%252F%252FDocuments%252FDemo%25
    2Epptx%26z%3D79AF7428%2D11BC%2D4D49%2DA0D8%2D4AC7DBA09A951" info="infoId" />
  <p:pres xmlns:p="http://schemas.microsoft.com/server/powerpoint/2009/mobile" ver="0">
    <p:docPr w="640" h="480" />
    <p:sldLst>
      <p:sld title="Selling a Product or Service" id="264" sldImg="img264.png"
        sldInfo="264.sldInfo.xml" />
      <p:sld title="Introduction" id="257" sldImg="img257.png" sldInfo="257.sldInfo.xml" />
      <p:sld title="Business opportunities" id="258" sldImg="img258.png"
        sldInfo="258.sldInfo.xml" />
      <p:sld title="Products and services " id="259" sldImg="img259.png"
        sldInfo="259.sldInfo.xml" />
      <p:sld title="Key Goals" id="262" sldImg="img262.png" sldInfo="262.sldInfo.xml" />
      <p:sld title="Cast comparison" id="260" sldImg="img260.png" sldInfo="260.sldInfo.xml"
        />
      <p:sld title="Next steps" id="263" sldImg="img263.png" sldInfo="263.sldInfo.xml" />
      <p:sld title="Conclusion" id="265" sldImg="img265.png" sldInfo="265.sldInfo.xml" />
    </p:sldLst>
    <p:showLst>
      <p:custShow name="Custom Show 1">
        <p:sld>264</p:sld>
        <p:sld>259</p:sld>
        <p:sld>263</p:sld>
      </p:custShow>
    </p:showLst>
    <p:showPr>
      <p:sldRg st="3" end="6" />
    </p:showPr>
  </p:pres>
</p:mobilePres>
```

The contained **slideImage** and **slideInfo** elements specify information how to retrieve the presentation slide images and the detailed information about a presentation slide respectively.

The **w** and **h** attributes specify that the width and height of the presentation are 640 pixels and 480 pixels respectively.

4.2 Slide Image

The **CT_SlideImage** ([\[MS-PWVRSC\]](#) section 2.3.10) element contained within **CT_MobilePres** ([\[MS-PWVRSC\]](#) section 2.3.12) element specifies how to retrieve the slide images.

The URL to retrieve the image of slide having identifier 264 of width 640 pixels and height 480 pixels is constructed as follows:

```
http://sptestamd/_layouts/MobileSlideImage.ashx?presentationId=d%3DH%253Asptestamd%252F%252FDocuments%252FDemo%252Epptx%26z%3DFE7280F5%2D1895%2D4F49%2DA3E8%2D8E4628A833E52&imageId=img264.png&width=640&height=480
```

The protocol client performs an HTTP GET on this URL to obtain the slide image.

4.3 Slide Information

The **CT_SlideInfo** ([\[MS-PWVRSC\]](#) section 2.3.11) element contained within a **CT_MobilePres** ([\[MS-PWVRSC\]](#) section 2.3.12) element specifies how to retrieve the slide information.

The URL to retrieve information about slide 264 is constructed as follows:

```
http://sptestamd/_layouts/MobileSlideInfo.ashx?presentationId=d%3DH%253Asptestamd%252F%252FDocuments%252FDemo%252Epptx%26z%3DFE7280F5%2D1895%2D4F49%2DA3E8%2D8E4628A833E52&infoId=264.sldInfo.xml
```

The protocol client performs an HTTP GET on this URL.

The protocol server responds with a **CT_Slide** ([\[MS-PWVRSC\]](#) section 2.3.18) element which contains information about the slide.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<p:sld xmlns:p="http://schemas.microsoft.com/server/powerpoint/2009/mobile" ver="0">
  <p:sldTxt>
    <p:ph type="t">
      <p:p>
        <p:t>Selling a Product or Service</p:t>
      </p:p>
    </p:ph>
  </p:sldTxt>
  <p:notes>
    <p:p buChar="•">
      <p:t>Total time: 30 minutes</p:t>
    </p:p>
    <p:p buChar="•">
      <p:t>Concentrate on Business opportunities</p:t>
    </p:p>
    <p:p level="2" buChar="•">
      <p:t>Spend 5 minutes on intro</p:t>
    </p:p>
  </p:notes>
</p:sld>
```

```
<p:p>
  <p:t/>
</p:p>
<p:p>
  <p:t/>
</p:p>
<p:p>
  <p:t>Text for test</p:t>
</p:p>
<p:p>
  <p:t/>
</p:p>
<p:p>
  <p:t/>
</p:p>
</p:notes>
</p:sld>
```

The contained **notes** element specifies the text content of the notes of a presentation slide.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

None.

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® FAST™ Search Server 2010
- Microsoft® PowerPoint® Mobile 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

This section identifies changes that were made to the [MS-OMPWHP] 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
Z Appendix B: Product Behavior	Updated the list of applicable products.	N	Content updated.

9 Index

A

Abstract data model
[server](#) 9
[Applicability](#) 7
[Attribute groups](#) 8
[Attributes](#) 8

C

[Capability negotiation](#) 7
[Change tracking](#) 20
Client
[overview](#) 9
[Complex types](#) 8

D

Data model - abstract
[server](#) 9

E

Events
[local - server](#) 13
[timer - server](#) 13
Examples
[overview](#) 14
[presentation information](#) 14
[slide image](#) 15
[slide information](#) 15

F

[Fields - vendor-extensible](#) 7
[Full WSDL](#) 18

G

[Glossary](#) 5
[Groups](#) 8

I

[Implementer - security considerations](#) 17
[Index of security parameters](#) 17
[Informative references](#) 6
Initialization
[server](#) 10
[Introduction](#) 5

L

Local events
[server](#) 13

M

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

N

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

O

Operations
[MobilePresentation](#) 10
[MobileSlideImage](#) 12
[MobileSlideInfo](#) 12
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 17
[Preconditions](#) 6
[Prerequisites](#) 6
[Presentation information example](#) 14
[Product behavior](#) 19

R

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

S

Security
[implementer considerations](#) 17
[parameter index](#) 17
Sequencing rules
[server](#) 10
Server
[abstract data model](#) 9
[details](#) 9
[initialization](#) 10
[local events](#) 13
[message processing](#) 10
[MobilePresentation operation](#) 10
[MobileSlideImage operation](#) 12
[MobileSlideInfo operation](#) 12

[overview](#) 9
[sequencing rules](#) 10
[timer events](#) 13
[timers](#) 9
[Simple types](#) 8
[Slide image example](#) 15
[Slide information example](#) 15
[Standards assignments](#) 7
Syntax
[messages - overview](#) 8

T

Timer events
[server](#) 13
Timers
[server](#) 9
[Tracking changes](#) 20
[Transport](#) 8
Types
[complex](#) 8
[simple](#) 8

V

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

W

[WSDL](#) 18