[MS-PWPHP]:

PowerPoint Web Presentation Handler Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
10/8/2012	1.0	New	Released new document.
2/11/2013	1.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.1	Minor	Clarified the meaning of the technical content.
11/18/2013	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.1	No Change	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	2.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The PowerPoint Web Presentation Handler Protocol 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 [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

horizontal alignment: A formatting setting that specifies how content is positioned within the horizontal space of a cell, object, or page. Content can be aligned along the left or right edge, or distributed evenly across the horizontal space.

horizontal indent: An indent that is used in a cell to adjust cell content horizontally.

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

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

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC4627]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

permission: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

presentation: A collection of slides that are intended to be viewed by an audience.

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

Request-URI: A URI in an **HTTP** request message, as described in [RFC2616].

website: A group of related webpages that is hosted by a server on the World Wide Web or an intranet. Each website has its own entry points, metadata, administration settings, and workflows. Also referred to as site.

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

1.2 References

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

1.2.1 Normative References

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

[ISO/IEC29500-1:2011] ISO/IEC, "Information Technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference", ISO/IEC 29500-1:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59575

[RFC1738] Berners-Lee, T., Masinter, L., and McCahill, M., Eds., "Uniform Resource Locators (URL)", RFC 1738, December 1994, http://www.ietf.org/rfc/rfc1738.txt

[RFC1945] Berners-Lee, T., Fielding, R., and Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, http://www.ietf.org/rfc/rfc1945.txt

[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

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, http://www.rfc-editor.org/rfc/rfc2396.txt

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.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

1.2.2 Informative References

None.

1.3 Overview

This protocol enables a protocol client to send a request to retrieve information about **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.

1.4 Relationship to Other Protocols

This protocol uses **HTTP** 1.0, as described in [RFC1945], HTTP 1.1, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818], for message transport.

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

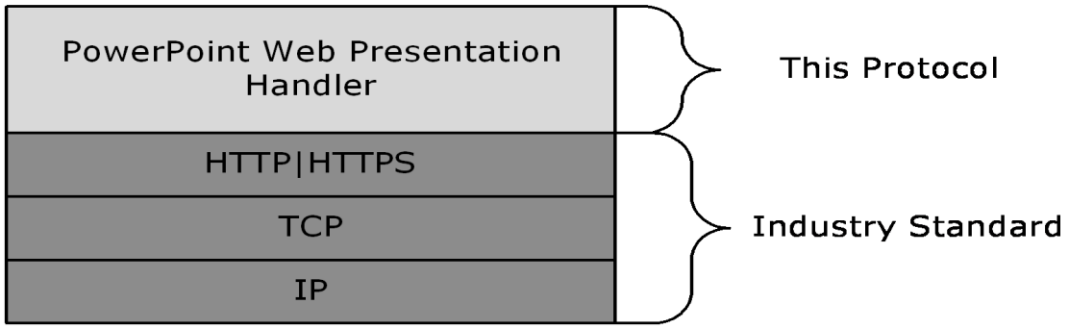

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **website** identified by a URL that is known by protocol clients. The protocol server endpoint is formed by appending /p/presentation.ashx to the URL of the website, for example: http://www.contoso.com/p/presentation.ashx.

The protocol assumes that authentication 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

Versioning and capability negotiation for this protocol is handled by either the HTTP protocols as described in [RFC1945] and [RFC2616], or the HTTPS protocol as described in [RFC2818].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

All protocol messages MUST use HTTP 1.0 (as specified in [RFC1945]), HTTP 1.1 (as specified in [RFC2616]), or the HTTPS protocol (as specified in [RFC2818]) for message transport. Additionally, protocol servers MUST support JSON [RFC4627] over HTTP and HTTPS.

2.2 Message Syntax

All messages in this protocol MUST be valid HTTP requests and responses as specified in [RFC2616].

2.2.1 Request Syntax

2.2.1.1 Request HTTP Version

The HTTP version MUST be either HTTP 1.0 or HTTP 1.1, as specified in [RFC2616] section 3.1.

2.2.1.2 Request HTTP Method

The protocol client MUST use the HTTP GET method as specified in [RFC2616] section 9.

2.2.1.3 Request-URI Syntax

The **Request-URI** MUST adhere to the following rules:

- The Request-URI sent in the HTTP request MUST be a valid URI as specified in [RFC1738].
- The Scheme Component MUST be either HTTP or HTTPS, as specified in [RFC2396] section 3.1.
- The protocol client MUST generate the Request-URI by appending "/p/presentation.ashx" to the website URI.

The Query Component of the Request-URI MUST be present and follow the rules specified in section 2.2.1.3.1 and [RFC2396] section 3.4.

2.2.1.3.1 Query Segment

The Query Component of the Request-URI MUST contain three query string parameters with the following names:

- pid
- ct
- wdSlideId

The value for each query string parameter MUST be a non-empty string and a valid query string parameter value as specified in [RFC2396]. The Query Component MUST NOT contain query string parameters other than the three required parameters, as specified in [RFC2396] section 3.4. The protocol server MUST NOT require the query string parameters to appear in any particular order.

The protocol client and protocol server MUST interpret the query string parameters as defined in the following table.

Parameter name	Description
pid	The identifier of the presentation to retrieve information about.
ct	The value for this parameter MUST be set to "slide".
wdSlideId	The identifier of the **presentation slide** to retrieve information about, as specified in [ISO/IEC29500-1:2011] section 19.7.13.

2.2.1.4 Request Header Syntax

The protocol client MUST provide valid Message Headers as specified in [RFC2616] section 4.2.

2.2.2 Response Syntax

2.2.2.1 Response Status

The protocol server MUST provide a valid status code as specified in [RFC2616] section 6.1.1.

2.2.2.2 Response Header Syntax

The protocol server MUST provide valid message headers as specified in [RFC2616] section 4.2.

2.2.2.3 Response Body Syntax

The protocol server MUST provide a valid message body as specified in [RFC2616] section 4.3.

The content of the message body MUST be a **JSON** array for **Slide** as specified in section 2.2.2.3.1.

2.2.2.3.1 Slide

A JSON object that specifies a presentation slide.

Members:

FHidden: A JSON true or false that specifies if the presentation slide is hidden.

Id: A JSON int that specifies the identifier of the presentation slide.

Notes: A JSON array of **CT_Paragraph**, as specified in section 2.2.2.3.2, that specifies the notes for the presentation slide.

Thumbnail: A JSON string that specifies the URL of the thumbnail for the presentation slide.

Title: A JSON string that specifies the title of the presentation slide.

2.2.2.3.2 CT_Paragraph

A JSON object that specifies a paragraph with simple format information.

Members:

t: A JSON string that specifies the concatenation of the text of all the runs of the paragraph.

level: A JSON int that specifies level of **horizontal indent**. It ranges from 0 to 255.

buChar: A JSON string that specifies the bullet.

align: An **ST_Alignment**, as specified in section 2.2.2.3.3, that specifies the **horizontal alignment** of text.

rtl: A JSON true or false that specifies whether the text goes from right to left.

2.2.2.3.3 ST_Alignment

This type specifies the horizontal alignment of text. Must be one of the following values:

Parameter name	Description
l	The text is left aligned.
c	The text is center aligned.
r	The text is right aligned.
j	The text is justified aligned.
d	The text is distributed aligned.

3 Protocol Details

3.1 Common Details

The protocol client sends an HTTP request to the protocol server as specified in section 2.2.1. The protocol server responds with an HTTP response as specified in section 2.2.2.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The message processing events and sequencing rules are as follows:

- The protocol client MUST generate a valid Request-URI as specified in section 2.2.1.3.
- The protocol client MUST issue the HTTP GET request to the generated URL.
- The protocol server MUST verify that the client has **permission** to access the specified URL and MUST return the appropriate Status Code as specified in [RFC2616] section 6.1.1 if the protocol client does not have permissions. The client MUST be prepared to accept all status codes specified in [RFC2616] section 6.1.1.
- After performing any implementation specific processing, the protocol server MUST return a valid HTTP response as specified in section 2.2.2.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following is an example of this protocol usage. In this example, the identifier of the presentation is the following:

```
http://www.contoso.com/documents/loremipsum.pptx
```

The URL generated by the protocol client to get information about slide 256 is as follows:

```
http://
www.contoso.com/p/presentation.ashx?Pid=http://www.contoso.com/documents/loremipsum.pptx&ct=s
lide&wdSlideId=256
```

HTTP request

```
GET http://
www.contoso.com/p/presentation.ashx?Pid=http://www.contoso.com/documents/loremipsum.pptx&ct=s
lide&wdSlideId=256 HTTP/1.1
Accept: */*
Host: contoso.com
Connection: Keep-Alive
```

HTTP response

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: application/json; charset=UTF-8
Expires: -1
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
P3P: CP="CAO DSP COR ADMa DEV CONi TELi CUR PSA PSD TAI IVDi OUR SAMi BUS DEM NAV STA UNI COM
INT PHY ONL FIN PUR"
X-Content-Type-Options: nosniff
X-Download-Options: noopen
Content-Disposition: attachment
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Fri, 07 Sep 2012 17:51:48 GMT
Content-Length: 1437

[{"FHidden":false,"Id":256,"Notes":[{"align":0,"buChar":"","level":1,"rtl":false,"t":"Lorem
ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in
voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est
laborum."}],"Thumbnail":".\/imagehandler.ashx?
Pid=http://www.contoso.com/documents/loremipsum.pptx&Rid=img256.png","Title":"Lorem ipsum
dolor sit amet"}]
```

5 Security

5.1 Security Considerations for Implementers

When using this protocol over an untrusted network, an HTTPS (as described in [RFC2818]) connection can help mitigate risks of protocol messages being intercepted or tampered with.

The information contained in the presentation identified by the pid query string parameter is likely to be security sensitive. For example, it could contain confidential data such as financial records. Therefore it is recommended that the protocol server establish that the protocol client has permissions to access the presentation.

There are no restrictions on the protocol server regarding the message header content type, as described in [RFC2616] section 14.17. Therefore, it is recommended that the protocol client checks the Content-Type to avoid running any executable file that could pose a security risk.

5.2 Index of Security Parameters

None.

6 Appendix A: 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 2013
- Microsoft PowerPoint 2016

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.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

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

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

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

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

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

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

8 Index