

[MS-PASCWS]:

PowerPoint Automation Services Conversion Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
1/20/2012	0.1	New	Released new document.
4/11/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	0.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	2.0	Major	Significantly changed the technical content.
7/15/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Overview	7
1.4	Relationship to Other Protocols	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
2	Messages.....	9
2.1	Transport	9
2.2	Common Message Syntax	9
2.2.1	Namespaces	9
2.2.2	Messages.....	10
2.2.2.1	StreamConversionRequest	10
2.2.2.2	StreamConversionResponse	10
2.2.3	Elements	11
2.2.4	Complex Types.....	11
2.2.4.1	ConversionSettings	11
2.2.4.2	FixedFormatSettings.....	11
2.2.4.3	PictureSettings	12
2.2.4.4	PresentationSettings	13
2.2.5	Simple Types	13
2.2.5.1	char	14
2.2.5.2	duration.....	14
2.2.5.3	guid	14
2.2.5.4	PictureFormat.....	14
2.2.5.5	PublishOption	15
2.2.5.6	ViewFormat.....	16
2.2.6	Attributes	18
2.2.7	Groups	18
2.2.8	Attribute Groups.....	18
3	Protocol Details	19
3.1	Server Details.....	19
3.1.1	Abstract Data Model.....	19
3.1.2	Timers	19
3.1.3	Initialization.....	20
3.1.4	Message Processing Events and Sequencing Rules	20
3.1.4.1	Convert	20
3.1.4.1.1	Messages	20
3.1.4.1.1.1	StreamConversionRequest	20
3.1.4.1.1.2	StreamConversionResponse	20
3.1.4.1.2	Elements.....	21
3.1.4.1.2.1	StreamConversionRequest	21
3.1.4.1.2.2	StreamConversionResponse	21
3.1.4.1.3	Complex Types	21
3.1.4.1.4	Simple Types	21
3.1.4.1.4.1	StreamBody	22
3.1.4.1.5	Attributes	22
3.1.4.1.6	Groups.....	22

3.1.4.1.7	Attribute Groups.....	22
3.1.5	Timer Events.....	22
3.1.6	Other Local Events.....	22
4	Protocol Examples.....	23
4.1	Convert a presentation to the PDF file format	23
5	Security.....	24
5.1	Security Considerations for Implementers	24
5.2	Index of Security Parameters	24
6	Appendix A: Full WSDL	25
7	Appendix B: Full XML Schema.....	27
7.1	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion Schema	27
7.2	http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface Schema	28
7.3	http://schemas.microsoft.com/2003/10/Serialization/ Schema	29
7.4	http://schemas.microsoft.com/Message Schema	30
7.5	http://tempuri.org/ Schema	30
8	Appendix C: Product Behavior	31
9	Change Tracking.....	32
10	Index.....	33

1 Introduction

The PowerPoint Automation Services Conversion Web Service Protocol enables a protocol client to convert presentations from one file format to another.

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

1.1 Glossary

This document uses the following terms:

certificate: A certificate is a collection of attributes (1) and extensions that can be stored persistently. The set of attributes in a certificate can vary depending on the intended usage of the certificate. A certificate securely binds a public key to the entity that holds the corresponding private key. A certificate is commonly used for authentication (2) and secure exchange of information on open networks, such as the Internet, extranets, and intranets. Certificates are digitally signed by the issuing certification authority (CA) and can be issued for a user, a computer, or a service. The most widely accepted format for certificates is defined by the ITU-T X.509 version 3 international standards. For more information about attributes and extensions, see [\[RFC3280\]](#) and [\[X509\]](#) sections 7 and 8.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [\[RFC4122\]](#) or [\[C706\]](#) must be used for generating the **GUID**. See also universally unique identifier (UUID).

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

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

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

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

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

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

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

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [\[SOAP1.2-1/2007\]](#) section 5.2 for more information.

SOAP message: An XML document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

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

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

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

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

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

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

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

1.2 References

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

1.2.1 Normative References

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

[ISO/IEC29500:2011] ISO/IEC, "Information technology -- Document description and processing languages -- Office Open XML File Formats -- Parts 1-4", ISO/IEC 29500-1:2011, 2011, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59575

[MS-PPTX] Microsoft Corporation, "[PowerPoint \(.pptx\) Extensions to the Office Open XML File Format](#)".

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

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

[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>

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

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

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

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

1.2.2 Informative References

[ISO-19005-1] International Organization for Standardization (ISO), "Document management -- Electronic document file format for long-term preservation -- Part 1: Use of PDF 1.4 (PDF/A-1)", 2005, http://www.iso.org/iso/catalogue_detail?csnumber=38920

Note There is a charge to download this item.

[JFIF] Hamilton, E., "JPEG File Interchange Format, Version 1.02", September 1992, <http://www.w3.org/Graphics/JPEG/jfif.txt>

[MSFT-XPS] Microsoft Corporation, "XML Paper Specification", Version 1.0, <http://www.microsoft.com/whdc/xps/xpsspec.mspx>

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

[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>

[W3C-PNG] World Wide Web Consortium, "Portable Network Graphics (PNG) Specification, Second Edition", November 2003, <http://www.w3.org/TR/PNG>

1.3 Overview

This protocol allows a protocol client to convert a **presentation** from one file format to another file format. It allows the protocol client to pass a presentation file to the protocol server and to receive from the protocol server a converted file in a different format. A typical scenario for using this protocol is a file conversion application that enables users to convert presentation files to a different format for archiving purposes.

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 [\[SOAP1.2/2\]](#). This protocol uses SOAP over **HTTP**, as described in [\[RFC2616\]](#), and SOAP over **HTTPS**, as described in [\[RFC2818\]](#), as shown in the following layering diagram.

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

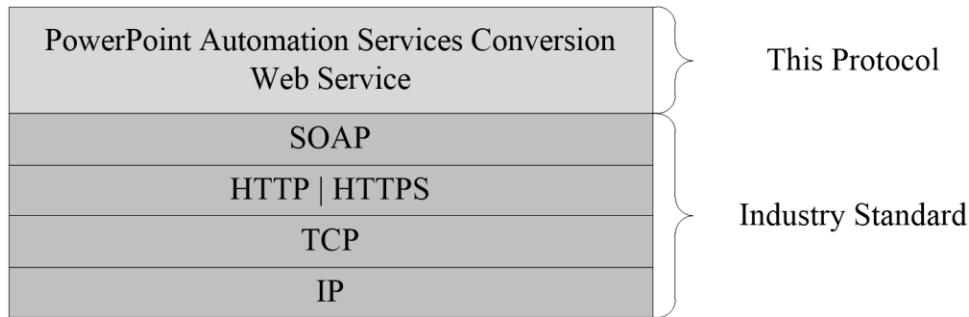


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a protocol server that exposes one or more endpoint (4) URIs that are known by protocol clients. The endpoint (4) URI of the protocol server and the transport that is used by the protocol server are either known by the protocol client or obtained by using the discovery mechanism that is described in [\[MS-SPTWS\]](#).

The protocol client obtains the requisite ApplicationClassId and ApplicationVersion values and the endpoint (4) URI of the protocol server that provides the discovery mechanism, as described in [\[MS-SPTWS\]](#), by means that are independent of either protocol.

This protocol requires the protocol client to have permission to call the methods on the protocol server.

The protocol client implements the token-based security mechanisms that are required by the protocol server and related security protocols, as described in [\[MS-SPSTWS\]](#).

1.6 Applicability Statement

This protocol is intended for use by protocol clients and protocol servers that are connected by high-bandwidth, low-latency network connections.

This protocol is designed for a protocol client to send a presentation file to a protocol server and get a converted file in a different format as response from the protocol server.

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 described in section [2.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

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

2.1 Transport

Protocol servers **MUST** support SOAP over HTTP or HTTPS. All protocol messages **MUST** be transported using HTTP bindings at the transport level.

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

If the HTTPS transport is used, a server **certificate** **MUST** be deployed.

This protocol does not define any means for activating a protocol server or protocol client. The protocol server **MUST** be configured and begin listening in an implementation-specific way. In addition, the protocol client **MUST** know the format and transport that is used by the protocol server; for example, the SOAP format over an HTTP transport.

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
q1	http://schemas.microsoft.com/Message	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://tempuri.org/	
tns1	http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion	
tns2	http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface	
tns3	http://schemas.microsoft.com/2003/10/Serialization/	
tns4	http://tempuri.org/Imports	
wsaw	http://www.w3.org/2006/05/addressing/wsdl	

Prefix	Namespace URI	Reference
wSDL	http://schemas.xmlsoap.org/wSDL/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] 1 [XMLSCHEMA2] 1

2.2.2 Messages

Message	Description
StreamConversionRequest	A SOAP request message used to convert a presentation from one file format to another file format.
StreamConversionResponse	A SOAP response message used to convert a presentation from one file format to another file format.

This specification does not define any common **WSDL message** definitions.

2.2.2.1 StreamConversionRequest

The **StreamConversionRequest** message is a SOAP request message used to convert a presentation from one file format to another file format.

- The **SOAP body** MUST be the **StreamConversionRequest** element as specified by section [3.1.4.1.2.1](#).
- The **SOAP header** MUST include a FileExtension element of type **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2 1) that specifies the extension of the input file.
- The SOAP header MUST include a Format element of type **ViewFormat** (section [2.2.5.6](#)) that specifies the file format of the output of the conversion operation.
- The SOAP header MUST include an Id element of type **guid** (section [2.2.5.3](#)) that specifies a unique identifier for the conversion operation.
- The SOAP header MUST include a Settings element of type **ConversionSettings** (section [2.2.4.1](#)) that specifies the output settings for the conversion operation.

2.2.2.2 StreamConversionResponse

The **StreamConversionResponse** message is a SOAP response message used to convert a presentation from one file format to another file format.

- The SOAP body MUST be the **StreamConversionResponse** element as specified by section [3.1.4.1.2.2](#).

- The SOAP header MUST include an `m_result` element of type **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the result of the conversion. A value of zero indicates the operation was successful. All other values indicate the operation was not successful.

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
ConversionSettings	A complex type that specifies output settings for a conversion operation. This type MUST NOT be used directly in an operation. Instead, an extension type MUST be used as specified by FixedFormatSettings (section 2.2.4.2), PictureSettings (section 2.2.4.3), or PresentationSettings (section 2.2.4.4).
FixedFormatSettings	A complex type that specifies output settings for a conversion operation that has a ViewFormat (section 2.2.5.6) value of PptPdf or Xps .
PictureSettings	A complex type that specifies output settings for a conversion operation that has a ViewFormat (section 2.2.5.6) value of PowerpointStaticView .
PresentationSettings	A complex type that specifies output settings for a conversion operation that has a ViewFormat (section 2.2.5.6) value of Pptx .

2.2.4.1 ConversionSettings

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion>

A complex type that specifies output settings for a conversion operation. This type MUST NOT be used directly in an operation. Instead, an extension type MUST be used as specified by **FixedFormatSettings** (section [2.2.4.2](#)), **PictureSettings** (section [2.2.4.3](#)), or **PresentationSettings** (section [2.2.4.4](#)).

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

2.2.4.2 FixedFormatSettings

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion>

A complex type that specifies output settings for a conversion operation that has a **ViewFormat** (section [2.2.5.6](#)) value of **PptPdf** or **Xps**.

```
<xs:complexType name="FixedFormatSettings" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tnsl:ConversionSettings">
```

```

<xs:sequence>
  <xs:element minOccurs="0" name="BitmapUnembeddableFonts" type="xs:boolean"/>
  <xs:element minOccurs="0" name="FrameSlides" type="xs:boolean"/>
  <xs:element minOccurs="0" name="IncludeDocumentProperties" type="xs:boolean"/>
  <xs:element minOccurs="0" name="IncludeDocumentStructureTags" type="xs:boolean"/>
  <xs:element minOccurs="0" name="IncludeHiddenSlides" type="xs:boolean"/>
  <xs:element minOccurs="0" name="OptimizeForMinimumSize" type="xs:boolean"/>
  <xs:element minOccurs="0" name="UsePdfA" type="xs:boolean"/>
  <xs:element minOccurs="0" name="UseVerticalOrder" type="xs:boolean"/>
  <xs:element minOccurs="0" name="m_endSlide" type="xs:unsignedInt"/>
  <xs:element minOccurs="0" name="m_publishOption" type="tns1:PublishOption"/>
  <xs:element minOccurs="0" name="m_startSlide" type="xs:unsignedInt"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

BitmapUnembeddableFonts: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether fonts that cannot be embedded in the output file are to be rasterized. **True** means fonts that cannot be embedded in the output file are rasterized.

FrameSlides: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether slides are to have a border. **True** means slides have a border.

IncludeDocumentProperties: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether document properties are to be included. **True** means document properties are included.

IncludeDocumentStructureTags: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether document structure tags are to be included. **True** means document structure tags are included.

IncludeHiddenSlides: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether hidden slides are to be included. **True** means hidden slides are included.

OptimizeForMinimumSize: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the output is to be optimized for minimum size. **True** means the output is optimized for minimum size.

UsePdfA: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the output file is to use the PDF/A format described in [ISO-19005-1]. **True** means the output file uses the PDF/A format described in [ISO-19005-1].

UseVerticalOrder: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the output is to use vertical ordering. **True** means the output uses vertical ordering.

m_endSlide: An **xs:unsignedInt** ([XMLSCHEMA2] section 3.3.22) that specifies the last slide to be included in the output.

m_publishOption: A **PublishOption** (section 2.2.5.5) that specifies the view.

m_startSlide: An **xs:unsignedInt** ([XMLSCHEMA2] section 3.3.22) that specifies the first slide to be included in the output.

2.2.4.3 PictureSettings

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion

A complex type that specifies output settings for a conversion operation that has a **ViewFormat** (section 2.2.5.6) value of **PowerpointStaticView**.

```

<xs:complexType name="PictureSettings" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tnsl:ConversionSettings">
      <xs:sequence>
        <xs:element minOccurs="0" name="m_height" type="xs:unsignedInt"/>
        <xs:element minOccurs="0" name="m_pictureFormat" type="tnsl:PictureFormat"/>
        <xs:element minOccurs="0" name="m_width" type="xs:unsignedInt"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

m_height: An **xs:unsignedInt** ([XMLSCHEMA2] section 3.3.22) that specifies the height of the image.

m_pictureFormat: A **PictureFormat** (section 2.2.5.4) that specifies the image format.

m_width: An **xs:unsignedInt** ([XMLSCHEMA2] section 3.3.22) that specifies the width of the image.

2.2.4.4 PresentationSettings

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion

A complex type that specifies output settings for a conversion operation that has a **ViewFormat** (section 2.2.5.6) value of **Pptx**.

```

<xs:complexType name="PresentationSettings" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexContent mixed="false">
    <xs:extension base="tnsl:ConversionSettings">
      <xs:sequence>
        <xs:element minOccurs="0" name="m_useStrict" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

m_useStrict: An **xs:boolean** ([XMLSCHEMA2] section 3.2.2) that specifies whether the output file is to adhere to the Strict conformance class as specified by [ISO/IEC29500:2011]. **True** means the output file adheres to the Strict conformance class as specified by [ISO/IEC29500:2011].

2.2.5 Simple Types

The following table summarizes the set of common XML schema simple type definitions defined by this specification. XML schema simple type definitions that are specific to a particular operation are described with the operation.

Simple type	Description
char	Reserved. The client and the server MUST NOT use this as the type of an element, and the client and the server MUST ignore it, if receiving an element of this type.
duration	Reserved. The client and the server MUST NOT use this as the type of an element, and the client and the server MUST ignore it, if receiving an element of this type.
guid	A string representing a GUID or unique identifier.
PictureFormat	A simple type that specifies an enumeration of image formats.

Simple type	Description
PublishOption	A simple type that specifies a view used when converting presentations to PDF or XPS file formats.
ViewFormat	A simple type that specifies the file format of the output of a conversion operation.

2.2.5.1 char

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

Reserved. The client and the server MUST NOT use this as the type of an element, and the client and the server MUST ignore it, if receiving an element of this type.

```
<xs:simpleType name="char" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
```

2.2.5.2 duration

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

Reserved. The client and the server MUST NOT use this as the type of an element, and the client and the server MUST ignore it, if receiving an element of this type.

```
<xs:simpleType name="duration" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.5.3 guid

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

A string representing a GUID or unique identifier.

```
<xs:simpleType name="guid" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.5.4 PictureFormat

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion>

A simple type that specifies an enumeration of image formats.

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

```

<xs:restriction base="xs:string">
  <xs:enumeration value="Default"/>
  <xs:enumeration value="Png"/>
  <xs:enumeration value="Jpg"/>
</xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for the **PictureFormat** simple type.

Value	Meaning
Default	The default format.
Png	Portable Network Graphics Format [W3C-PNG] .
Jpg	JPEG File Interchange Format [JFIF] .

2.2.5.5 PublishOption

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion>

A simple type that specifies a view used when converting presentations to PDF or XPS file formats.

```

<xs:simpleType name="PublishOption" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Default"/>
    <xs:enumeration value="Slides"/>
    <xs:enumeration value="Outline"/>
    <xs:enumeration value="Handout1"/>
    <xs:enumeration value="Handout2"/>
    <xs:enumeration value="Handout3"/>
    <xs:enumeration value="Handout4"/>
    <xs:enumeration value="Handout6"/>
    <xs:enumeration value="Handout9"/>
  </xs:restriction>
</xs:simpleType>

```

The following table specifies the allowable values for the **PublishOption** simple type.

Value	Meaning
Default	The default option.
Slides	Slides view.
Outline	Outline view.
Handout1	Handout view with one slide per page.
Handout2	Handout view with two slides per page.
Handout3	Handout view with three slides per page.
Handout4	Handout view with four slides per page.
Handout6	Handout view with six slides per page.

Value	Meaning
Handout9	Handout view with nine slides per page.

2.2.5.6 ViewFormat

Namespace:

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface>

A simple type that specifies the file format of the output of a conversion operation.

```
<xs:simpleType name="ViewFormat" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Invalid"/>
    <xs:enumeration value="Silverlight"/>
    <xs:enumeration value="Png"/>
    <xs:enumeration value="Pdf"/>
    <xs:enumeration value="Xps"/>
    <xs:enumeration value="Docx"/>
    <xs:enumeration value="Docm"/>
    <xs:enumeration value="Doc"/>
    <xs:enumeration value="Mht"/>
    <xs:enumeration value="Rtf"/>
    <xs:enumeration value="Xml"/>
    <xs:enumeration value="WordMobileImage"/>
    <xs:enumeration value="PowerpointSlideShow"/>
    <xs:enumeration value="PowerpointReadingView"/>
    <xs:enumeration value="PowerpointStaticView"/>
    <xs:enumeration value="Pptx"/>
    <xs:enumeration value="AccessiblePdf"/>
    <xs:enumeration value="AutoPrintPdf"/>
    <xs:enumeration value="PptPdf"/>
    <xs:enumeration value="Odt"/>
    <xs:enumeration value="WordEdit"/>
    <xs:enumeration value="PowerPointSmall"/>
    <xs:enumeration value="WordTeaser"/>
    <xs:enumeration value="MobileBrowserPng"/>
    <xs:enumeration value="PowerPointExtraSmall"/>
    <xs:enumeration value="PowerPointExtraLarge"/>
    <xs:enumeration value="PowerPointStaticLarge"/>
    <xs:enumeration value="PowerPointMedia"/>
    <xs:enumeration value="MontageView"/>
    <xs:enumeration value="PowerPointExtraExtraLarge"/>
    <xs:enumeration value="WordThumbnail"/>
    <xs:enumeration value="PowerPointStaticSmall"/>
    <xs:enumeration value="DocxStrict"/>
    <xs:enumeration value="Ppsx"/>
    <xs:enumeration value="Potx"/>
    <xs:enumeration value="PptxStrict"/>
    <xs:enumeration value="ReaderDocx"/>
    <xs:enumeration value="ReaderXml"/>
    <xs:enumeration value="PlaceholderDeprecatedPngSmall"/>
    <xs:enumeration value="PngLarge"/>
    <xs:enumeration value="PageMetadata"/>
    <xs:enumeration value="WordOoxml"/>
    <xs:enumeration value="Odp"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for the **ViewFormat** simple type.

Value	Meaning
Invalid	Reserved. MUST be ignored.
Silverlight	Reserved. MUST be ignored.
Png	Reserved. MUST be ignored.
Pdf	Reserved. MUST be ignored.
Xps	XML Paper Specification as described by [MSFT-XPS] .
Docx	Reserved. MUST be ignored.
Docm	Reserved. MUST be ignored.
Doc	Reserved. MUST be ignored.
Mht	Reserved. MUST be ignored.
Rtf	Reserved. MUST be ignored.
Xml	Reserved. MUST be ignored.
WordMobileImage	Reserved. MUST be ignored.
PowerpointSlideShow	Reserved. MUST be ignored.
PowerpointReadingView	Reserved. MUST be ignored.
PowerpointStaticView	Image file.
Pptx	Office OpenXML PresentationML as specified by [ISO/IEC29500:2011] and [MS-PPTX] .
AccessiblePdf	Reserved. MUST be ignored.
AutoPrintPdf	Reserved. MUST be ignored.
PptPdf	Portable Document Format as described by [ISO-19005-1] .
Odt	Reserved. MUST be ignored.
WordEdit	Reserved. MUST be ignored.
PowerPointSmall	Reserved. MUST be ignored.
WordTeaser	Reserved. MUST be ignored.
MobileBrowserPng	Reserved. MUST be ignored.
PowerPointExtraSmall	Reserved. MUST be ignored.
PowerPointExtraLarge	Reserved. MUST be ignored.
PowerPointStaticLarge	Reserved. MUST be ignored.
PowerPointMedia	Reserved. MUST be ignored.
MontageView	Reserved. MUST be ignored.
PowerPointExtraExtraLarge	Reserved. MUST be ignored.
WordThumbnail	Reserved. MUST be ignored.

Value	Meaning
PowerPointStaticSmall	Reserved. MUST be ignored.
DocxStrict	Reserved. MUST be ignored.
Ppsx	Reserved. MUST be ignored.
Potx	Reserved. MUST be ignored.
PptxStrict	Reserved. MUST be ignored.
ReaderDocx	
ReaderXml	
PlaceholderDeprecatedPngSmall	
PngLarge	
PageMetadata	
WordOoxml	
Odp	

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

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

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

This protocol allows protocol servers to notify protocol clients of application-level faults by using SOAP faults. Except where otherwise specified, these SOAP faults are not significant for interoperability and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and to notify protocol clients of authorization faults by using either HTTP status codes or SOAP faults, as specified previously in this section.

3.1 Server Details

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

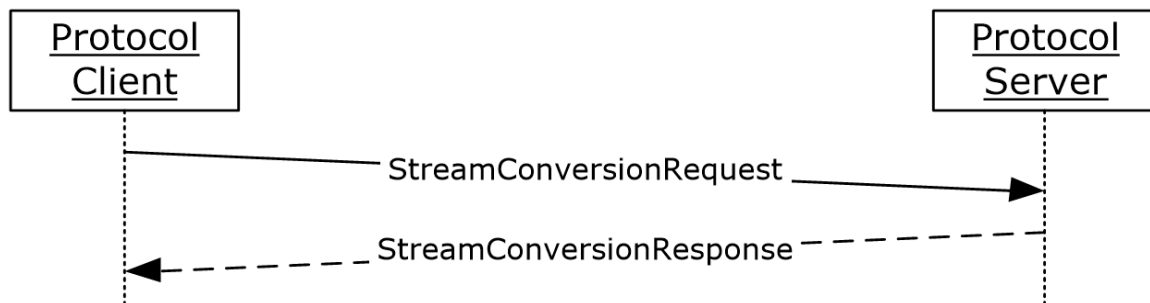


Figure 2: Sample communication between protocol client and protocol server

The protocol client sends the **StreamConversionRequest SOAP message** as specified in section [3.1.4.1.1.1](#) to convert a presentation. The protocol server responds with a **StreamConversionResponse** as specified in section [3.1.4.1.1.2](#), which contains the converted presentation.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
Convert	The Convert operation is used to convert a presentation from one file format to another file format.

3.1.4.1 Convert

The **Convert** operation is used to convert a presentation from one file format to another file format.

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

```
<wsdl:operation name="Convert" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input wsaw:Action="http://tempuri.org/IConversionService/Convert"
name="StreamConversionRequest" message="tns:StreamConversionRequest"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output wsaw:Action="http://tempuri.org/IConversionService/ConvertResponse"
name="StreamConversionResponse" message="tns:StreamConversionResponse"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends a **StreamConversionRequest** request message, and the protocol server responds with a **StreamConversionResponse** response message.

3.1.4.1.1 Messages

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

Message	Description
StreamConversionRequest	The request WSDL message for the Convert WSDL operation.
StreamConversionResponse	The response WSDL message for the Convert WSDL operation.

3.1.4.1.1.1 StreamConversionRequest

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

The **SOAP action** value is:

```
http://tempuri.org/IConversionService/Convert
```

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

3.1.4.1.1.2 StreamConversionResponse

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

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

3.1.4.1.2 Elements

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

Element	Description
StreamConversionRequest	The input data for the Convert WSDL operation.
StreamConversionResponse	The result data for the Convert WSDL operation.

3.1.4.1.2.1 StreamConversionRequest

The **StreamConversionRequest** element specifies the input data for the **Convert** WSDL operation.

```
<xs:element name="StreamConversionRequest" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q1="http://schemas.microsoft.com/Message" name="InputData"
type="q1:StreamBody"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

InputData: A **StreamBody** (section [3.1.4.1.4.1](#)) element that specifies the contents of the input presentation file.

3.1.4.1.2.2 StreamConversionResponse

The **StreamConversionResponse** element specifies the result data for the **Convert** WSDL operation.

```
<xs:element name="StreamConversionResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q1="http://schemas.microsoft.com/Message" name="m_outputData"
type="q1:StreamBody"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

m_outputData: A **StreamBody** (section [3.1.4.1.4.1](#)) element that specifies the contents of the converted output file.

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

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

Simple type	Description
StreamBody	A stream of bytes.

3.1.4.1.4.1 StreamBody

Namespace: http://schemas.microsoft.com/Message

A simple type that specifies a stream of bytes.

```
<xs:simpleType name="StreamBody" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:restriction base="xs:base64Binary"/>  
</xs:simpleType>
```

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

4.1 Convert a presentation to the PDF file format

This example demonstrates how a protocol client converts a presentation to the PDF file format.

The protocol client sends the following **StreamConversionRequest** message to the protocol server.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <FileExtension>pptx</FileExtension>
    <Format>PptPdf</Format>
    <Id>936DA01F-9ABD-4d9d-80C7-02AF85C822A8</Id>
    <Settings>
      <a:BitmapUnembeddableFonts>true</a:BitmapUnembeddableFonts>
      <a:FrameSlides>true</a:FrameSlides>
      <a:IncludeComments>true</a:IncludeComments>
      <a:IncludeDocumentProperties>true</a:IncludeDocumentProperties>
      <a:IncludeDocumentStructureTags>true</a:IncludeDocumentStructureTags>
      <a:IncludeHiddenSlides>true</a:IncludeHiddenSlides>
      <a:OptimizeForMinimumSize>false</a:OptimizeForMinimumSize>
      <a:UsePdfA>false</a:UsePdfA>
      <a:UseVerticalOrder>false</a:UseVerticalOrder>
      <a:m endSlide>0</a:m endSlide>
      <a:m_publishOption>Slides</a:m_publishOption>
      <a:m startSlide>0</a:m startSlide>
    </Settings>
  </s:Header>
  <s:Body>
    <StreamConversionRequest
      xmlns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion"
    >
      <InputData>aaaaa...</InputData>
    </StreamConversionRequest>
  </s:Body>
</s:Envelope>
```

The protocol server responds with the following **StreamConversionResponse** message to the protocol client.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <m_result>0</m_result>
  </s:Header>
  <s:Body>
    <StreamConversionResponse
      xmlns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion"
    >
      <m_outputData>bbbbbb...</m_outputData>
    </StreamConversionResponse>
  </s:Body>
</s:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://tempuri.org/"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema xmlns:tns4="http://tempuri.org/Imports"
targetNamespace="http://tempuri.org/Imports">
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Convers
ion"/>
      <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInte
rface"/>
      <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/">
      <xs:import namespace="http://schemas.microsoft.com/Message/">
      <xs:import namespace="http://tempuri.org/">
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="IConversionService">
    <wsdl:operation name="Convert">
      <wsdl:input wsaw:Action="http://tempuri.org/IConversionService/Convert"
name="StreamConversionRequest" message="tns:StreamConversionRequest"/>
      <wsdl:output wsaw:Action="http://tempuri.org/IConversionService/ConvertResponse"
name="StreamConversionResponse" message="tns:StreamConversionResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DefaultBinding_IConversionService" type="tns:IConversionService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Convert">
      <soap:operation soapAction="http://tempuri.org/IConversionService/Convert"
style="document"/>
      <wsdl:input name="StreamConversionRequest">
        <soap:header message="tns:StreamConversionRequest_Headers" part="FileExtension"
use="literal"/>
        <soap:header message="tns:StreamConversionRequest_Headers" part="Format"
use="literal"/>
        <soap:header message="tns:StreamConversionRequest_Headers" part="Id" use="literal"/>
        <soap:header message="tns:StreamConversionRequest_Headers" part="Settings"
use="literal"/>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="StreamConversionResponse">
        <soap:header message="tns:StreamConversionResponse_Headers" part="m_result"
use="literal"/>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:message name="StreamConversionRequest">
    <wsdl:part name="parameters" element="tns:StreamConversionRequest"/>
  </wsdl:message>
  <wsdl:message name="StreamConversionRequest_Headers">
    <wsdl:part name="FileExtension" element="tns:FileExtension"/>
    <wsdl:part name="Format" element="tns:Format"/>
    <wsdl:part name="Id" element="tns:Id"/>
    <wsdl:part name="Settings" element="tns:Settings"/>
  </wsdl:message>
  <wsdl:message name="StreamConversionResponse">
    <wsdl:part name="parameters" element="tns:StreamConversionResponse"/>
  </wsdl:message>
  <wsdl:message name="StreamConversionResponse_Headers">
    <wsdl:part name="m_result" element="tns:m_result"/>
  </wsdl:message>

```

```
</wsdl:message>  
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

Schema name	Prefix	Section
http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion	tns1	7.1
http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface	tns2	7.2
http://schemas.microsoft.com/2003/10/Serialization/	tns3	7.3
http://schemas.microsoft.com/Message	q1	7.4
http://tempuri.org/	tns	7.5

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

7.1 <http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion> Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion" elementFormDefault="qualified"
  targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
  <xs:complexType name="PictureSettings">
    <xs:complexContent mixed="false">
      <xs:extension base="tns1:ConversionSettings">
        <xs:sequence>
          <xs:element minOccurs="0" name="m_height" type="xs:unsignedInt"/>
          <xs:element minOccurs="0" name="m_pictureFormat" type="tns1:PictureFormat"/>
          <xs:element minOccurs="0" name="m_width" type="xs:unsignedInt"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="PictureSettings" nillable="true" type="tns1:PictureSettings"/>
  <xs:complexType name="ConversionSettings">
    <xs:sequence/>
  </xs:complexType>
  <xs:element name="ConversionSettings" nillable="true" type="tns1:ConversionSettings"/>
  <xs:simpleType name="PictureFormat">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Default"/>
      <xs:enumeration value="Png"/>
      <xs:enumeration value="Jpg"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="PictureFormat" nillable="true" type="tns1:PictureFormat"/>
  <xs:complexType name="PresentationSettings">
    <xs:complexContent mixed="false">
      <xs:extension base="tns1:ConversionSettings">
        <xs:sequence>
          <xs:element minOccurs="0" name="m_useStrict" type="xs:boolean"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="PresentationSettings" nillable="true" type="tns1:PresentationSettings"/>
  <xs:complexType name="FixedFormatSettings">
```

```

<xs:complexContent mixed="false">
  <xs:extension base="tns1:ConversionSettings">
    <xs:sequence>
      <xs:element minOccurs="0" name="BitmapUnembeddableFonts" type="xs:boolean"/>
      <xs:element minOccurs="0" name="FrameSlides" type="xs:boolean"/>
      <xs:element minOccurs="0" name="IncludeDocumentProperties" type="xs:boolean"/>
      <xs:element minOccurs="0" name="IncludeDocumentStructureTags" type="xs:boolean"/>
      <xs:element minOccurs="0" name="IncludeHiddenSlides" type="xs:boolean"/>
      <xs:element minOccurs="0" name="OptimizeForMinimumSize" type="xs:boolean"/>
      <xs:element minOccurs="0" name="UsePdfA" type="xs:boolean"/>
      <xs:element minOccurs="0" name="UseVerticalOrder" type="xs:boolean"/>
      <xs:element minOccurs="0" name="m_endSlide" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="m_publishOption" type="tns1:PublishOption"/>
      <xs:element minOccurs="0" name="m_startSlide" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="FixedFormatSettings" nillable="true" type="tns1:FixedFormatSettings"/>
<xs:simpleType name="PublishOption">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Default"/>
    <xs:enumeration value="Slides"/>
    <xs:enumeration value="Outline"/>
    <xs:enumeration value="Handout1"/>
    <xs:enumeration value="Handout2"/>
    <xs:enumeration value="Handout3"/>
    <xs:enumeration value="Handout4"/>
    <xs:enumeration value="Handout6"/>
    <xs:enumeration value="Handout9"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="PublishOption" nillable="true" type="tns1:PublishOption"/>
</xs:schema>

```

7.2 <http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface> Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface" elementFormDefault="qualified"
  targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="ViewFormat">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Invalid"/>
      <xs:enumeration value="Silverlight"/>
      <xs:enumeration value="Png"/>
      <xs:enumeration value="Pdf"/>
      <xs:enumeration value="Xps"/>
      <xs:enumeration value="Docx"/>
      <xs:enumeration value="Docm"/>
      <xs:enumeration value="Doc"/>
      <xs:enumeration value="Mht"/>
      <xs:enumeration value="Rtf"/>
      <xs:enumeration value="Xml"/>
      <xs:enumeration value="WordMobileImage"/>
      <xs:enumeration value="PowerpointSlideShow"/>
      <xs:enumeration value="PowerpointReadingView"/>
      <xs:enumeration value="PowerpointStaticView"/>
      <xs:enumeration value="Pptx"/>
      <xs:enumeration value="AccessiblePdf"/>
      <xs:enumeration value="AutoPrintPdf"/>
      <xs:enumeration value="PptPdf"/>
      <xs:enumeration value="Odt"/>
      <xs:enumeration value="WordEdit"/>
    </xs:restriction>
  </xs:simpleType>

```

```

    <xs:enumeration value="PowerPointSmall"/>
    <xs:enumeration value="WordTeaser"/>
    <xs:enumeration value="MobileBrowserPng"/>
    <xs:enumeration value="PowerPointExtraSmall"/>
    <xs:enumeration value="PowerPointExtraLarge"/>
    <xs:enumeration value="PowerPointStaticLarge"/>
    <xs:enumeration value="PowerPointMedia"/>
    <xs:enumeration value="MontageView"/>
    <xs:enumeration value="PowerPointExtraExtraLarge"/>
    <xs:enumeration value="WordThumbnail"/>
    <xs:enumeration value="PowerPointStaticSmall"/>
    <xs:enumeration value="DocxStrict"/>
    <xs:enumeration value="Ppsx"/>
    <xs:enumeration value="Potx"/>
    <xs:enumeration value="PptxStrict"/>
    <xs:enumeration value="ReaderDocx"/>
    <xs:enumeration value="ReaderXml"/>
    <xs:enumeration value="PlaceholderDeprecatedPngSmall"/>
    <xs:enumeration value="PngLarge"/>
    <xs:enumeration value="PageMetadata"/>
    <xs:enumeration value="WordOoxml"/>
    <xs:enumeration value="Odp"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="ViewFormat" nillable="true" type="tns2:ViewFormat"/>
</xs:schema>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns3="http://schemas.microsoft.com/2003/10/Serialization/"
  attributeFormDefault="qualified" elementFormDefault="qualified"
  targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary" nillable="true" type="xs:base64Binary"/>
  <xs:element name="boolean" nillable="true" type="xs:boolean"/>
  <xs:element name="byte" nillable="true" type="xs:byte"/>
  <xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
  <xs:element name="decimal" nillable="true" type="xs:decimal"/>
  <xs:element name="double" nillable="true" type="xs:double"/>
  <xs:element name="float" nillable="true" type="xs:float"/>
  <xs:element name="int" nillable="true" type="xs:int"/>
  <xs:element name="long" nillable="true" type="xs:long"/>
  <xs:element name="QName" nillable="true" type="xs:QName"/>
  <xs:element name="short" nillable="true" type="xs:short"/>
  <xs:element name="string" nillable="true" type="xs:string"/>
  <xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
  <xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
  <xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
  <xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>
  <xs:element name="char" nillable="true" type="tns3:char"/>
  <xs:simpleType name="char">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns3:duration"/>
  <xs:simpleType name="duration">
    <xs:restriction base="xs:duration">
      <xs:pattern value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)S)?)?"/>
      <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
      <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="guid" nillable="true" type="tns3:guid"/>
  <xs:simpleType name="guid">
    <xs:restriction base="xs:string">

```

```

        <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}"/>
    </xs:restriction>
</xs:simpleType>
<xs:attribute name="FactoryType" type="xs:QName"/>
<xs:attribute name="Id" type="xs:ID"/>
<xs:attribute name="Ref" type="xs:IDREF"/>
</xs:schema>

```

7.4 http://schemas.microsoft.com/Message Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/Message"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="StreamBody">
        <xs:restriction base="xs:base64Binary"/>
    </xs:simpleType>
</xs:schema>

```

7.5 http://tempuri.org/ Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="StreamConversionRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element xmlns:q1="http://schemas.microsoft.com/Message" name="InputData"
type="q1:StreamBody"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="FileExtension" nillable="true" type="xs:string"/>
    <xs:element
xmlns:tns2="http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInt
erface" name="Format" type="tns2:ViewFormat"/>
    <xs:element xmlns:tns3="http://schemas.microsoft.com/2003/10/Serialization/" name="Id"
type="tns3:guid"/>
    <xs:element
xmlns:tns1="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conver
sion" name="Settings" nillable="true" type="tns1:ConversionSettings"/>
    <xs:element name="StreamConversionResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element xmlns:q1="http://schemas.microsoft.com/Message" name="m outputData"
type="q1:StreamBody"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="m_result" type="xs:int"/>
    <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Convers
ion"/>
    <xs:import
namespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInte
rface"/>
    <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
    <xs:import namespace="http://schemas.microsoft.com/Message"/>
</xs:schema>

```

8 Appendix C: Product Behavior

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

- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 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.

9 Change Tracking

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

10 Index

A

Abstract data model
[server](#) 19
[Applicability](#) 8
[Attribute groups](#) 18
[Attributes](#) 18

C

[Capability negotiation](#) 8
[Change tracking](#) 32
[char simple type](#) 14
Client
[overview](#) 19
[Complex types](#) 11
[ConversionSettings](#) 11
[FixedFormatSettings](#) 11
[PictureSettings](#) 12
[PresentationSettings](#) 13
[ConversionSettings complex type](#) 11
[Convert a presentation to the PDF file format example](#) 23

D

Data model - abstract
[server](#) 19
[duration simple type](#) 14

E

Events
[local - server](#) 22
[timer - server](#) 22
Example
[convert a presentation to the PDF file format](#) 23

F

[Fields - vendor-extensible](#) 8
[FixedFormatSettings complex type](#) 11
[Full WSDL](#) 25
[Full XML schema](#) 27

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion.Schema> 27

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface.Schema> 28

<http://schemas.microsoft.com/2003/10/Serialization/Schema> 29
<http://schemas.microsoft.com/Message.Schema> 30
<http://tempuri.org/Schema> 30

G

[Glossary](#) 5

[Groups](#) 18
[guid simple type](#) 14

I

[Implementer - security considerations](#) 24
[Index of security parameters](#) 24
[Informative references](#) 7
Initialization
[server](#) 20
[Introduction](#) 5

L

Local events
[server](#) 22

M

Message processing
[server](#) 20
Messages
[attribute groups](#) 18
[attributes](#) 18
[char simple type](#) 14
[complex types](#) 11
[ConversionSettings complex type](#) 11
[duration simple type](#) 14
[elements](#) 11
[enumerated](#) 10
[FixedFormatSettings complex type](#) 11
[groups](#) 18
[guid simple type](#) 14
[namespaces](#) 9
[PictureFormat simple type](#) 14
[PictureSettings complex type](#) 12
[PresentationSettings complex type](#) 13
[PublishOption simple type](#) 15
[simple types](#) 13
[StreamConversionRequest](#) 10
[StreamConversionRequest message](#) 10
[StreamConversionResponse](#) 10
[StreamConversionResponse message](#) 10
[syntax](#) 9
[transport](#) 9
[ViewFormat simple type](#) 16

N

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

O

Operations
[Convert](#) 20
[Overview \(synopsis\)](#) 7

P

[Parameters - security index](#) 24
[PictureFormat simple type](#) 14

[PictureSettings complex type](#) 12
[Preconditions](#) 8
[Prerequisites](#) 8
[PresentationSettings complex type](#) 13
[Product behavior](#) 31
Protocol Details
 [overview](#) 19
[PublishOption simple type](#) 15

R

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

S

Security
 [implementer considerations](#) 24
 [parameter index](#) 24
Sequencing rules
 [server](#) 20
Server
 [abstract data model](#) 19
 [Convert operation](#) 20
 [details](#) 19
 [initialization](#) 20
 [local events](#) 22
 [message processing](#) 20
 [overview](#) 19
 [sequencing rules](#) 20
 [timer events](#) 22
 [timers](#) 19
[Simple types](#) 13
 [char](#) 14
 [duration](#) 14
 [guid](#) 14
 [PictureFormat](#) 14
 [PublishOption](#) 15
 [ViewFormat](#) 16
[Standards assignments](#) 8
Syntax
 [messages - overview](#) 9

T

Timer events
 [server](#) 22
Timers
 [server](#) 19
[Tracking changes](#) 32
[Transport](#) 9
Types
 [complex](#) 11
 [simple](#) 13

V

[Vendor-extensible fields](#) 8
[Versioning](#) 8
[ViewFormat simple type](#) 16

W

[WSDL](#) 25

X

[XML schema](#) 27

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.PowerPoint.Conversion.Schema> 27

<http://schemas.datacontract.org/2004/07/Microsoft.Office.Web.Conversion.ViewerInterface.Schema> 28

<http://schemas.microsoft.com/2003/10/Serialization/Schema> 29

<http://schemas.microsoft.com/Message.Schema> 30

<http://tempuri.org/Schema> 30