

[MS-PWEDPS]: PowerPoint Web Editor Data 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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). 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 iplq@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	Minor	Clarified the meaning of 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	2.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	11
1.1	Glossary	11
1.2	References.....	11
1.2.1	Normative References	11
1.2.2	Informative References	12
1.3	Protocol Overview (Synopsis)	13
1.4	Relationship to Other Protocols.....	13
1.5	Prerequisites/Preconditions	13
1.6	Applicability Statement.....	14
1.7	Versioning and Capability Negotiation.....	14
1.8	Vendor-Extensible Fields.....	14
1.9	Standards Assignments	14
2	Messages.....	15
2.1	Transport.....	15
2.2	Common Message Syntax	15
2.2.1	Namespaces	15
2.2.2	Messages	15
2.2.3	Elements.....	16
2.2.4	Complex Types	16
2.2.4.1	ArrayOfContentMasterFe.....	18
2.2.4.2	ArrayOfEditShape	19
2.2.4.3	ArrayOfEditSlide	19
2.2.4.4	ArrayOfEditSlideChanges	19
2.2.4.5	ArrayOfEditSlideInfoFe	19
2.2.4.6	ArrayOfMainMasterFe	20
2.2.4.7	ArrayOfunsignedInt.....	20
2.2.4.8	ArrayOfViewElement	20
2.2.4.9	ClippingInfo	21
2.2.4.10	ContentMasterFe.....	21
2.2.4.11	EditCommandResponse	21
2.2.4.12	EditPresentationChanges	22
2.2.4.13	EditPresentationInfo	23
2.2.4.14	EditShape	24
2.2.4.15	EditSlide	25
2.2.4.16	EditSlideChanges	26
2.2.4.17	EditSlideInfoFe	26
2.2.4.18	FontSizeMapping.....	27
2.2.4.19	Html.....	27
2.2.4.20	MainMasterFe	28
2.2.4.21	Notes	28
2.2.4.22	Point	28
2.2.4.23	PptViewingService.PrintResult	29
2.2.4.24	Rectangle	29
2.2.4.25	ServiceError.....	30
2.2.4.26	ServiceResult	30
2.2.4.27	SlideIdList.....	30
2.2.4.28	SmartArtInfo	31
2.2.4.29	Theme.....	31
2.2.4.30	ThemeColor	32

2.2.4.31 ViewElement	32
2.2.5 Simple Types	33
2.2.5.1 char	34
2.2.5.2 ClientActions	34
2.2.5.3 duration	35
2.2.5.4 guid	35
2.2.5.5 ServiceErrorType	35
2.2.6 Attributes	36
2.2.7 Groups	36
2.2.8 Attribute Groups	36
3 Protocol Details	37
3.1 Server Details	37
3.1.1 Abstract Data Model	38
3.1.2 Timers	38
3.1.3 Initialization	38
3.1.4 Message Processing Events and Sequencing Rules	38
3.1.4.1 ApplyTheme	39
3.1.4.1.1 Messages	40
3.1.4.1.1.1 IPptEditInternal_ApplyTheme_InputMessage	40
3.1.4.1.1.2 IPptEditInternal_ApplyTheme_OutputMessage	40
3.1.4.1.2 Elements	40
3.1.4.1.2.1 ApplyTheme	40
3.1.4.1.2.2 ApplyThemeResponse	41
3.1.4.1.3 Complex Types	41
3.1.4.1.4 Simple Types	41
3.1.4.1.5 Attributes	41
3.1.4.1.6 Groups	41
3.1.4.1.7 Attribute Groups	41
3.1.4.2 ChangePictureStyle	41
3.1.4.2.1 Messages	42
3.1.4.2.1.1 IPptEditInternal_ChangePictureStyle_InputMessage	42
3.1.4.2.1.2 IPptEditInternal_ChangePictureStyle_OutputMessage	42
3.1.4.2.2 Elements	42
3.1.4.2.2.1 ChangePictureStyle	42
3.1.4.2.2.2 ChangePictureStyleResponse	43
3.1.4.2.3 Complex Types	43
3.1.4.2.4 Simple Types	43
3.1.4.2.5 Attributes	43
3.1.4.2.6 Groups	43
3.1.4.2.7 Attribute Groups	43
3.1.4.3 ChangeSmartArtColor	43
3.1.4.3.1 Messages	44
3.1.4.3.1.1 IPptEditInternal_ChangeSmartArtColor_InputMessage	44
3.1.4.3.1.2 IPptEditInternal_ChangeSmartArtColor_OutputMessage	44
3.1.4.3.2 Elements	44
3.1.4.3.2.1 ChangeSmartArtColor	44
3.1.4.3.2.2 ChangeSmartArtColorResponse	45
3.1.4.3.3 Complex Types	45
3.1.4.3.4 Simple Types	45
3.1.4.3.5 Attributes	45
3.1.4.3.6 Groups	45
3.1.4.3.7 Attribute Groups	45

3.1.4.4	ChangeSmartArtLayout	45
3.1.4.4.1	Messages	46
3.1.4.4.1.1	IPptEditInternal_ChangeSmartArtLayout_InputMessage	46
3.1.4.4.1.2	IPptEditInternal_ChangeSmartArtLayout_OutputMessage	46
3.1.4.4.2	Elements	46
3.1.4.4.2.1	ChangeSmartArtLayout	46
3.1.4.4.2.2	ChangeSmartArtLayoutResponse	47
3.1.4.4.3	Complex Types	47
3.1.4.4.4	Simple Types	47
3.1.4.4.5	Attributes	47
3.1.4.4.6	Groups	47
3.1.4.4.7	Attribute Groups	47
3.1.4.5	ChangeSmartArtStyle	47
3.1.4.5.1	Messages	48
3.1.4.5.1.1	IPptEditInternal_ChangeSmartArtStyle_InputMessage	48
3.1.4.5.1.2	IPptEditInternal_ChangeSmartArtStyle_OutputMessage	48
3.1.4.5.2	Elements	48
3.1.4.5.2.1	ChangeSmartArtStyle	48
3.1.4.5.2.2	ChangeSmartArtStyleResponse	49
3.1.4.5.3	Complex Types	49
3.1.4.5.4	Simple Types	49
3.1.4.5.5	Attributes	49
3.1.4.5.6	Groups	49
3.1.4.5.7	Attribute Groups	49
3.1.4.6	ClearPlaceholder	49
3.1.4.6.1	Messages	50
3.1.4.6.1.1	IPptEditInternal_ClearPlaceholder_InputMessage	50
3.1.4.6.1.2	IPptEditInternal_ClearPlaceholder_OutputMessage	50
3.1.4.6.2	Elements	50
3.1.4.6.2.1	ClearPlaceholder	50
3.1.4.6.2.2	ClearPlaceholderResponse	51
3.1.4.6.3	Complex Types	51
3.1.4.6.4	Simple Types	51
3.1.4.6.5	Attributes	51
3.1.4.6.6	Groups	51
3.1.4.6.7	Attribute Groups	51
3.1.4.7	DeleteSlide	51
3.1.4.7.1	Messages	52
3.1.4.7.1.1	IPptEditInternal_DeleteSlide_InputMessage	52
3.1.4.7.1.2	IPptEditInternal_DeleteSlide_OutputMessage	52
3.1.4.7.2	Elements	52
3.1.4.7.2.1	DeleteSlide	52
3.1.4.7.2.2	DeleteSlideResponse	53
3.1.4.7.3	Complex Types	53
3.1.4.7.4	Simple Types	53
3.1.4.7.5	Attributes	53
3.1.4.7.6	Groups	53
3.1.4.7.7	Attribute Groups	53
3.1.4.8	DuplicateSlide	53
3.1.4.8.1	Messages	54
3.1.4.8.1.1	IPptEditInternal_DuplicateSlide_InputMessage	54
3.1.4.8.1.2	IPptEditInternal_DuplicateSlide_OutputMessage	54
3.1.4.8.2	Elements	54

3.1.4.8.2.1 DuplicateSlide	54
3.1.4.8.2.2 DuplicateSlideResponse	55
3.1.4.8.3 Complex Types	55
3.1.4.8.4 Simple Types	55
3.1.4.8.5 Attributes	55
3.1.4.8.6 Groups	55
3.1.4.8.7 Attribute Groups	55
3.1.4.9 GetEditPresInfo	55
3.1.4.9.1 Messages	56
3.1.4.9.1.1 IPptEditInternal_GetEditPresInfo_InputMessage	56
3.1.4.9.1.2 IPptEditInternal_GetEditPresInfo_OutputMessage	56
3.1.4.9.2 Elements	56
3.1.4.9.2.1 GetEditPresInfo	56
3.1.4.9.2.2 GetEditPresInfoResponse	57
3.1.4.9.3 Complex Types	57
3.1.4.9.4 Simple Types	57
3.1.4.9.5 Attributes	57
3.1.4.9.6 Groups	57
3.1.4.9.7 Attribute Groups	57
3.1.4.10 GetEditSlide	57
3.1.4.10.1 Messages	58
3.1.4.10.1.1 IPptEditInternal_GetEditSlide_InputMessage	58
3.1.4.10.1.2 IPptEditInternal_GetEditSlide_OutputMessage	58
3.1.4.10.2 Elements	58
3.1.4.10.2.1 GetEditSlide	58
3.1.4.10.2.2 GetEditSlideResponse	58
3.1.4.10.3 Complex Types	59
3.1.4.10.4 Simple Types	59
3.1.4.10.5 Attributes	59
3.1.4.10.6 Groups	59
3.1.4.10.7 Attribute Groups	59
3.1.4.11 GetPresentationId	59
3.1.4.11.1 Messages	59
3.1.4.11.1.1 IPptEditInternal_GetPresentationId_InputMessage	59
3.1.4.11.1.2 IPptEditInternal_GetPresentationId_OutputMessage	60
3.1.4.11.2 Elements	60
3.1.4.11.2.1 GetPresentationId	60
3.1.4.11.2.2 GetPresentationIdResponse	60
3.1.4.11.3 Complex Types	60
3.1.4.11.4 Simple Types	61
3.1.4.11.5 Attributes	61
3.1.4.11.6 Groups	61
3.1.4.11.7 Attribute Groups	61
3.1.4.12 InsertClipart	61
3.1.4.12.1 Messages	61
3.1.4.12.1.1 IPptEditInternal_InsertClipart_InputMessage	61
3.1.4.12.1.2 IPptEditInternal_InsertClipart_OutputMessage	61
3.1.4.12.2 Elements	62
3.1.4.12.2.1 InsertClipart	62
3.1.4.12.2.2 InsertClipartResponse	62
3.1.4.12.3 Complex Types	63
3.1.4.12.4 Simple Types	63
3.1.4.12.5 Attributes	63

3.1.4.12.6	Groups.....	63
3.1.4.12.7	Attribute Groups.....	63
3.1.4.13	InsertSlide	63
3.1.4.13.1	Messages	63
3.1.4.13.1.1	IPptEditInternal_InsertSlide_InputMessage	63
3.1.4.13.1.2	IPptEditInternal_InsertSlide_OutputMessage	64
3.1.4.13.2	Elements	64
3.1.4.13.2.1	InsertSlide	64
3.1.4.13.2.2	InsertSlideResponse.....	64
3.1.4.13.3	Complex Types	65
3.1.4.13.4	Simple Types	65
3.1.4.13.5	Attributes	65
3.1.4.13.6	Groups.....	65
3.1.4.13.7	Attribute Groups.....	65
3.1.4.14	InsertSmartArt	65
3.1.4.14.1	Messages	65
3.1.4.14.1.1	IPptEditInternal_InsertSmartArt_InputMessage	65
3.1.4.14.1.2	IPptEditInternal_InsertSmartArt_OutputMessage	66
3.1.4.14.2	Elements	66
3.1.4.14.2.1	InsertSmartArt	66
3.1.4.14.2.2	InsertSmartArtResponse.....	66
3.1.4.14.3	Complex Types	67
3.1.4.14.4	Simple Types	67
3.1.4.14.5	Attributes	67
3.1.4.14.6	Groups.....	67
3.1.4.14.7	Attribute Groups.....	67
3.1.4.15	MoveSlide	67
3.1.4.15.1	Messages	67
3.1.4.15.1.1	IPptEditInternal_MoveSlide_InputMessage	67
3.1.4.15.1.2	IPptEditInternal_MoveSlide_OutputMessage	68
3.1.4.15.2	Elements	68
3.1.4.15.2.1	MoveSlide	68
3.1.4.15.2.2	MoveSlideResponse.....	68
3.1.4.15.3	Complex Types	69
3.1.4.15.4	Simple Types	69
3.1.4.15.5	Attributes	69
3.1.4.15.6	Groups.....	69
3.1.4.15.7	Attribute Groups.....	69
3.1.4.16	Noop	69
3.1.4.16.1	Messages	69
3.1.4.16.1.1	IPptEditInternal_Noop_InputMessage	69
3.1.4.16.1.2	IPptEditInternal_Noop_OutputMessage.....	70
3.1.4.16.2	Elements	70
3.1.4.16.2.1	Noop	70
3.1.4.16.2.2	NoopResponse.....	70
3.1.4.16.3	Complex Types	70
3.1.4.16.4	Simple Types	70
3.1.4.16.5	Attributes	70
3.1.4.16.6	Groups.....	70
3.1.4.16.7	Attribute Groups.....	71
3.1.4.17	Print.....	71
3.1.4.17.1	Messages	71
3.1.4.17.1.1	IPptEditInternal_Print_InputMessage.....	71

3.1.4.17.1.2	IPptEditInternal_Print_OutputMessage	71
3.1.4.17.2	Elements	71
3.1.4.17.2.1	Print.....	71
3.1.4.17.2.2	PrintResponse	72
3.1.4.17.3	Complex Types	72
3.1.4.17.4	Simple Types	72
3.1.4.17.5	Attributes	72
3.1.4.17.6	Groups.....	72
3.1.4.17.7	Attribute Groups.....	72
3.1.4.18	Redo	72
3.1.4.18.1	Messages	73
3.1.4.18.1.1	IPptEditInternal_Redo_InputMessage	73
3.1.4.18.1.2	IPptEditInternal_Redo_OutputMessage	73
3.1.4.18.2	Elements	73
3.1.4.18.2.1	Redo	73
3.1.4.18.2.2	RedoResponse	73
3.1.4.18.3	Complex Types	74
3.1.4.18.4	Simple Types	74
3.1.4.18.5	Attributes	74
3.1.4.18.6	Groups.....	74
3.1.4.18.7	Attribute Groups.....	74
3.1.4.19	ReplaceNotes	74
3.1.4.19.1	Messages	75
3.1.4.19.1.1	IPptEditInternal_ReplaceNotes_InputMessage	75
3.1.4.19.1.2	IPptEditInternal_ReplaceNotes_OutputMessage	75
3.1.4.19.2	Elements	75
3.1.4.19.2.1	ReplaceNotes	75
3.1.4.19.2.2	ReplaceNotesResponse	75
3.1.4.19.3	Complex Types	76
3.1.4.19.4	Simple Types	76
3.1.4.19.5	Attributes	76
3.1.4.19.6	Groups.....	76
3.1.4.19.7	Attribute Groups.....	76
3.1.4.20	ReplaceText	76
3.1.4.20.1	Messages	76
3.1.4.20.1.1	IPptEditInternal_ReplaceText_InputMessage	77
3.1.4.20.1.2	IPptEditInternal_ReplaceText_OutputMessage	77
3.1.4.20.2	Elements	77
3.1.4.20.2.1	ReplaceText	77
3.1.4.20.2.2	ReplaceTextResponse	77
3.1.4.20.3	Complex Types	78
3.1.4.20.4	Simple Types	78
3.1.4.20.5	Attributes	78
3.1.4.20.6	Groups.....	78
3.1.4.20.7	Attribute Groups.....	78
3.1.4.21	ResetSmartArt.....	78
3.1.4.21.1	Messages	78
3.1.4.21.1.1	IPptEditInternal_ResetSmartArt_InputMessage	79
3.1.4.21.1.2	IPptEditInternal_ResetSmartArt_OutputMessage	79
3.1.4.21.2	Elements	79
3.1.4.21.2.1	ResetSmartArt.....	79
3.1.4.21.2.2	ResetSmartArtResponse	79
3.1.4.21.3	Complex Types	80

3.1.4.21.4	Simple Types	80
3.1.4.21.5	Attributes	80
3.1.4.21.6	Groups	80
3.1.4.21.7	Attribute Groups	80
3.1.4.22	ReverseSmartArt	80
3.1.4.22.1	Messages	80
3.1.4.22.1.1	IPptEditInternal_ReverseSmartArt_InputMessage	80
3.1.4.22.1.2	IPptEditInternal_ReverseSmartArt_OutputMessage	81
3.1.4.22.2	Elements	81
3.1.4.22.2.1	ReverseSmartArt	81
3.1.4.22.2.2	ReverseSmartArtResponse	81
3.1.4.22.3	Complex Types	82
3.1.4.22.4	Simple Types	82
3.1.4.22.5	Attributes	82
3.1.4.22.6	Groups	82
3.1.4.22.7	Attribute Groups	82
3.1.4.23	SaveAndClose	82
3.1.4.23.1	Messages	82
3.1.4.23.1.1	IPptEditInternal_SaveAndClose_InputMessage	82
3.1.4.23.1.2	IPptEditInternal_SaveAndClose_OutputMessage	83
3.1.4.23.2	Elements	83
3.1.4.23.2.1	SaveAndClose	83
3.1.4.23.2.2	SaveAndCloseResponse	83
3.1.4.23.3	Complex Types	84
3.1.4.23.4	Simple Types	84
3.1.4.23.5	Attributes	84
3.1.4.23.6	Groups	84
3.1.4.23.7	Attribute Groups	84
3.1.4.24	ShowHideSlide	84
3.1.4.24.1	Messages	84
3.1.4.24.1.1	IPptEditInternal_ShowHideSlide_InputMessage	84
3.1.4.24.1.2	IPptEditInternal_ShowHideSlide_OutputMessage	85
3.1.4.24.2	Elements	85
3.1.4.24.2.1	ShowHideSlide	85
3.1.4.24.2.2	ShowHideSlideResponse	85
3.1.4.24.3	Complex Types	85
3.1.4.24.4	Simple Types	86
3.1.4.24.5	Attributes	86
3.1.4.24.6	Groups	86
3.1.4.24.7	Attribute Groups	86
3.1.4.25	Undo	86
3.1.4.25.1	Messages	86
3.1.4.25.1.1	IPptEditInternal_Undo_InputMessage	86
3.1.4.25.1.2	IPptEditInternal_Undo_OutputMessage	86
3.1.4.25.2	Elements	87
3.1.4.25.2.1	Undo	87
3.1.4.25.2.2	UndoResponse	87
3.1.4.25.3	Complex Types	87
3.1.4.25.4	Simple Types	87
3.1.4.25.5	Attributes	87
3.1.4.25.6	Groups	87
3.1.4.25.7	Attribute Groups	87
3.1.4.26	UpdateCanary	88

3.1.4.26.1	Messages	88
3.1.4.26.1.1	IPptEditInternal_UpdateCanary_InputMessage	88
3.1.4.26.1.2	IPptEditInternal_UpdateCanary_OutputMessage	88
3.1.4.26.2	Elements	88
3.1.4.26.2.1	UpdateCanary	88
3.1.4.26.2.2	UpdateCanaryResponse	89
3.1.4.26.3	Complex Types	89
3.1.4.26.4	Simple Types	89
3.1.4.26.5	Attributes	89
3.1.4.26.6	Groups.....	89
3.1.4.26.7	Attribute Groups.....	89
3.1.4.27	WaitCrash	89
3.1.4.27.1	Messages	90
3.1.4.27.1.1	IPptEditInternal_WaitCrash_InputMessage	90
3.1.4.27.1.2	IPptEditInternal_WaitCrash_OutputMessage.....	90
3.1.4.27.2	Elements	90
3.1.4.27.2.1	WaitCrash	90
3.1.4.27.2.2	WaitCrashResponse	90
3.1.4.27.3	Complex Types	91
3.1.4.27.4	Simple Types	91
3.1.4.27.5	Attributes	91
3.1.4.27.6	Groups.....	91
3.1.4.27.7	Attribute Groups.....	91
3.1.5	Timer Events	91
3.1.6	Other Local Events	91
4	Protocol Examples.....	92
4.1	Sample Protocol Interaction	92
5	Security.....	103
5.1	Security Considerations for Implementers.....	103
5.2	Index of Security Parameters	103
6	Appendix A: Full WSDL.....	104
7	Appendix B: Product Behavior.....	126
8	Change Tracking.....	128
9	Index	130

1 Introduction

This document specifies the PowerPoint Web Editor Data Protocol, which enables a protocol client to obtain information about and modify slide collection content on a protocol server.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
language code identifier (LCID)

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

absolute URL
canary
cookie
cross-site request forgery
cross-site scripting
Hypertext Markup Language (HTML)
main master slide
placeholder
presentation
presentation slide
right-to-left
shape
Simple Object Access Protocol (SOAP)
slide layout
SmartArt
SOAP action
SOAP body
SOAP fault
SOAP message
theme
title master slide
Uniform Resource Locator (URL)
Web Services Description Language (WSDL)
Web site
WSDL message
WSDL operation
XML namespace

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

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.

[ISO/IEC-29500-1] International Organization for Standardization, "Information Technology - Document description and processing languages - Office Open XML File Formats - Part 1: Fundamentals and Markup Language Reference", ISO/IEC PRF 29500-1, 2008, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51463

[MS-LCID] Microsoft Corporation, "[Windows Language Code Identifier \(LCID\) Reference](#)".

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

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

[RFC4627] Crockford, D., "The application/json Media Type for Javascript Object Notation (JSON)", RFC 4627, July 2006, <http://www.ietf.org/rfc/rfc4627.txt>

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

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

[WSA1.0] World Wide Web Consortium, "Web Services Addressing 1.0 - WSDL Binding", W3C Candidate Recommendation, May 2006, <http://www.w3.org/TR/2006/CR-ws-addr-wsdl-20060529/>

[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

[HTML] World Wide Web Consortium, "HTML 4.01 Specification", December 1999, <http://www.w3.org/TR/html4/>

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

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

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

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

[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>

1.3 Protocol Overview (Synopsis)

This protocol enables a protocol client to send a request to modify **presentation** content to the protocol server and then receive from the protocol server information about the result of the modification.

To facilitate this, the protocol allows the protocol client to send a request to the protocol server and then receive from the protocol server information about the existence of **presentation slides**, **main master slides**, and **slide layouts**. The protocol client can also send a request to the protocol server and then receive from the protocol server images of slide content.

1.4 Relationship to Other Protocols

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

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

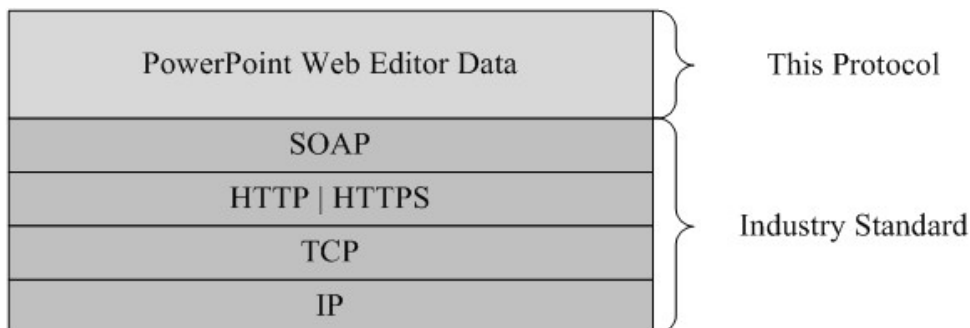


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a set of **Web sites (1)** that is identified by a set of **URLs** that are known by protocol clients.

The protocol server endpoint to which the client sends requests to modify presentation content is formed by appending `"/_vti_bin/ppt/edit.svc"` to the URL of the Web site (1), for example `http://www.contoso.com/Repository/_vti_bin/ppt/edit.svc`.

The protocol server endpoint to which the client sends requests to modify image content is formed by appending `"/_layouts/pptInsertPicture.ashx"` to the URL of the Web site (1), for example `http://www.contoso.com/Repository/_layouts/pptInsertPicture.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 and to modify presentations that are stored on 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**, **present**, and **not null**.

2.1 Transport

Protocol servers **MUST** support SOAP over HTTP. Protocol servers **SHOULD** additionally support SOAP over HTTPS for securing communication with clients. Protocol servers **SHOULD** support JSON [\[RFC4627\]](#) over HTTP and HTTPS. [<1>](#)

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
tns2	http://schemas.microsoft.com/2003/10/Serialization/Arrays	
tns1	http://schemas.datacontract.org/2004/07/p	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
wsaw	http://www.w3.org/2006/05/addressing/wsdl	[WSA1.0]
tns	http://tempuri.org/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2 Messages

None.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
ArrayOfContentMasterFe	A complex type that specifies a list of ContentMasterFe elements.
ArrayOfEditShape	A complex type that specifies a list of EditShape elements.
ArrayOfEditSlide	A complex type that specifies a list of EditSlide elements.
ArrayOfEditSlideChanges	A complex type that specifies a list of EditSlideChanges elements.
ArrayOfEditSlideInfoFe	A complex type that specifies a list of EditSlideInfoFe elements.
ArrayOfMainMasterFe	A complex type that specifies a list of MainMasterFe elements.
ArrayOfViewElement	A complex type that specifies a list of ViewElement elements.
ContentMasterFe	A complex type that specifies a slide layout.
EditCommandResponse	A complex type that specifies the response of a modification to a presentation.
EditPresentationChanges	A complex type that specifies the content of the presentation that is modified as a result of a request to the protocol server.
EditPresentationInfo	A complex type that specifies information about the content of a specified presentation.
EditShape	A complex type that specifies a shape.
EditSlide	A complex type that specifies a presentation slide.
EditSlideChanges	A complex type that specifies the content of the presentation slide that is modified as a result of a request to the protocol server.
EditSlideInfoFe	A complex type that specifies information about a presentation slide.
FontSizeMapping	Reserved. MUST be ignored.
Html	A complex type that specifies HTML content.
MainMasterFe	A complex type that specifies a main master slide.
Notes	A complex type that specifies the notes of a presentation slide.
Point	A complex type that specifies a two dimensional point.
Rectangle	A complex type that specifies a rectangle.
ServiceError	A complex type that specifies error information returned by the protocol server to a protocol client.

Complex type	Description
ServiceResult	A complex type that specifies the result of a protocol method. The protocol server returns this type to the protocol client containing either a successful Result element or an Error element.
SlideIdList	A complex type that specifies a list of identifiers for presentation slides.
SmartArtInfo	A complex type that specifies information about a SmartArt diagram.
Theme	A complex type that specifies information about a theme .
ThemeColor	A complex type that specifies a color using the red, green, blue RGB color model. Red, green, and blue are each expressed as an unsigned byte from 0 for no intensity to 255 for full intensity.
ViewElement	A complex type that specifies a visual representation of presentation slide content.
ArrayOfunsignedInt	A complex type that specifies a list of xs:unsignedInt ([XMLSCHEMA2] section 3.3.22) elements.

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
ArrayOfContentMasterFe	A complex type that specifies a list of ContentMasterFe elements.
ArrayOfEditShape	A complex type that specifies a list of EditShape elements.
ArrayOfEditSlide	A complex type that specifies a list of EditSlide elements.
ArrayOfEditSlideChanges	A complex type that specifies a list of EditSlideChanges elements.
ArrayOfEditSlideInfoFe	A complex type that specifies a list of EditSlideInfoFe elements.
ArrayOfMainMasterFe	A complex type that specifies a list of MainMasterFe elements.
ArrayOfunsignedInt	A complex type that specifies a list of xs:unsignedInt elements ([XMLSCHEMA2] section 3.3.22).
ArrayOfViewElement	A complex type that specifies a list of ViewElement elements.
ClippingInfo	A complex type that SHOULD <2> specify a region within an image.
ContentMasterFe	A complex type that specifies a slide layout.
EditCommandResponse	A complex type that specifies the response of a modification to a presentation.
EditPresentationChanges	A complex type that specifies the content of the presentation that is modified as a result of a request to the protocol server.
EditPresentationInfo	A complex type that specifies information about the content of a specified presentation.
EditShape	A complex type that specifies a shape.

Complex Type	Description
EditSlide	A complex type that specifies a presentation slide.
EditSlideChanges	A complex type that specifies the content of the presentation slide that is modified as a result of a request to the protocol server.
EditSlideInfoFe	A complex type that specifies information about a presentation slide.
FontSizeMapping	Reserved. MUST be ignored.
Html	A complex type that specifies HTML content.
MainMasterFe	A complex type that specifies a main master slide.
Notes	A complex type that specifies the notes of a presentation slide.
Point	A complex type that specifies a two dimensional point.
PptViewingService.PrintResult	A complex type that SHOULD §3 specify the result of a print operation.
Rectangle	A complex type that specifies a rectangle.
ServiceError	A complex type that specifies error information returned by the protocol server to a protocol client.
ServiceResult	A complex type that specifies the result of a protocol method. The protocol server returns this type to the protocol client containing either a successful Result element or an Error element.
SlideIdList	A complex type that specifies a list of identifiers for presentation slides.
SmartArtInfo	A complex type that specifies information about a SmartArt diagram.
Theme	A complex type that specifies information about a theme.
ThemeColor	A complex type that specifies a color using the red, green, blue RGB color model. Red, green, and blue are each expressed as an unsigned byte from 0 for no intensity to 255 for full intensity.
ViewElement	A complex type that specifies a visual representation of presentation slide content.

2.2.4.1 ArrayOfContentMasterFe

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a list of [ContentMasterFe](#) elements.

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

ContentMasterFe: Each element MUST specify a ContentMasterFe type instance.

2.2.4.2 ArrayOfEditShape

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [EditShape](#) elements.

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

EditShape: Each element MUST specify an EditShape type instance.

2.2.4.3 ArrayOfEditSlide

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [EditSlide](#) elements.

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

EditSlide: Each element MUST specify an EditSlide type instance.

2.2.4.4 ArrayOfEditSlideChanges

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [EditSlideChanges](#) elements.

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

EditSlideChanges: Each element MUST specify an EditSlideChanges type instance.

2.2.4.5 ArrayOfEditSlideInfoFe

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [EditSlideInfoFe](#) elements.

```
<xs:complexType name="ArrayOfEditSlideInfoFe">
  <xs:sequence>
```

```

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

```

EditSlideInfoFe: Each element MUST specify an EditSlideInfoFe type instance.

2.2.4.6 ArrayOfMainMasterFe

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [MainMasterFe](#) elements.

```

<xs:complexType name="ArrayOfMainMasterFe">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="MainMasterFe" nillable="true"
type="tns:MainMasterFe"/>
  </xs:sequence>
</xs:complexType>

```

MainMasterFe: Each element MUST specify a MainMasterFe type instance.

2.2.4.7 ArrayOfunsignedInt

Namespace: http://schemas.microsoft.com/2003/10/Serialization/Arrays

A complex type that specifies a list of **xs:unsignedInt** elements ([\[XMLSCHEMA2\]](#) section 3.3.22).

```

<xs:complexType name="ArrayOfunsignedInt">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="unsignedInt"
type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>

```

unsignedInt: Each element MUST specify an **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22).

2.2.4.8 ArrayOfViewElement

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of [ViewElement](#) elements.

```

<xs:complexType name="ArrayOfViewElement">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ViewElement" nillable="true"
type="tns:ViewElement"/>
  </xs:sequence>
</xs:complexType>

```

ViewElement: Each element MUST specify a ViewElement type instance.

2.2.4.9 ClippingInfo

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that SHOULD [4](#) specify a region within an image.

```
<xs:complexType name="ClippingInfo">
  <xs:sequence>
    <xs:element minOccurs="0" name="Bounds" nillable="true" type="tns:Rectangle"/>
    <xs:element minOccurs="0" name="TotalHeight" type="xs:int"/>
    <xs:element minOccurs="0" name="TotalWidth" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

Bounds: A **Rectangle** (section [2.2.4.24](#)) that specifies the bounds of the region. This element MUST be present.

TotalHeight: An **xs:int** ([XMLSCHEMA2](#) section 3.3.17) that specifies the total height of the image.

TotalWidth: An **xs:int** ([XMLSCHEMA2](#) section 3.3.17) that specifies the total width of the image.

2.2.4.10 ContentMasterFe

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a slide layout.

```
<xs:complexType name="ContentMasterFe">
  <xs:sequence>
    <xs:element minOccurs="0" name="Id" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="SlideLayoutType" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Id: An **xs:unsignedInt** ([XMLSCHEMA2](#) section 3.3.22) that specifies the identifier for the slide layout. This element MUST be present.

Name: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) that specifies the name of the slide layout.

SlideLayoutType: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) that specifies the layout type as specified by **ST_SlideLayoutType** [ISO/IEC-29500-1](#) section 19.7.15. This element MUST be present.

2.2.4.11 EditCommandResponse

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies the response of a modification to a presentation.

```
<xs:complexType name="EditCommandResponse">
  <xs:sequence>
    <xs:element minOccurs="0" name="PresChanges" nillable="true"
      type="tns:EditPresentationChanges"/>
  </xs:sequence>
</xs:complexType>
```

```

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

```

PresChanges: An [EditPresentationChanges](#) that specifies the content of the presentation that is modified.

2.2.4.12 EditPresentationChanges

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies the content of the presentation that is modified as a result of a request to the protocol server.

```

<xs:complexType name="EditPresentationChanges">
  <xs:sequence>
    <xs:element minOccurs="0" name="AddedSlideInfos" nillable="true"
type="tns:ArrayOfEditSlideInfoFe"/>
    <xs:element minOccurs="0" name="AddedSlides" nillable="true"
type="tns:ArrayOfEditSlide"/>
    <xs:element minOccurs="0" name="InvalidateAllThumbs" type="xs:boolean"/>
    <xs:element minOccurs="0" name="MainMasterIds" nillable="true" type="tns:SlideIdList"/>
    <xs:element minOccurs="0" name="ModifiedSlides" nillable="true"
type="tns:ArrayOfEditSlideChanges"/>
    <xs:element minOccurs="0" name="Redo" type="xs:boolean"/>
    <xs:element minOccurs="0" name="SlideOrder" nillable="true" type="tns:SlideIdList"/>
    <xs:element minOccurs="0" name="UndidApplyTheme" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Undo" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>

```

AddedSlideInfos: An **ArrayOfEditSlideInfoFe** element (section [2.2.4.5](#)) that specifies information about presentation slides added to the presentation. This element **MUST** be present.

AddedSlides: An **ArrayOfEditSlide** element (section [2.2.4.3](#)) that specifies presentation slides added to the presentation. This element **MUST** be present.

InvalidateAllThumbs: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that **SHOULD** [<5>](#) specify whether all thumbnails are invalid. This element **MUST** be present.

MainMasterIds: A **SlideIdList** element (section [2.2.4.27](#)) that specifies main master slides that are referenced by presentation slides.

ModifiedSlides: An **ArrayOfEditSlideChanges** element (section [2.2.4.4](#)) that specifies slide content that is modified. **MUST** be present.

Redo: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether **Redo** (section [3.1.4.18](#)) is possible. This element **MUST** be present.

SlideOrder: A **SlideIdList** that specifies the order of the presentation slides.

UndidApplyTheme: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that **SHOULD** [<6>](#) specify whether an applied theme was removed. This element **MUST** be present.

Undo: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether **Undo** (section [3.1.4.25](#)) is possible. This element **MUST** be present.

2.2.4.13 EditPresentationInfo

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies information about the content of a specified presentation.

```
<xs:complexType name="EditPresentationInfo">
  <xs:sequence>
    <xs:element minOccurs="0" name="CompatProblems" type="xs:boolean"/>
    <xs:element minOccurs="0" name="EditSlideInfos" nillable="true"
type="tns:ArrayOfEditSlideInfoFe"/>
    <xs:element minOccurs="0" name="FCreateNew" type="xs:boolean"/>
    <xs:element minOccurs="0" name="FirstSlideIndex" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="HasComments" type="xs:boolean"/>
    <xs:element minOccurs="0" name="MainMasters" nillable="true"
type="tns:ArrayOfMainMasterFe"/>
    <xs:element minOccurs="0" name="Redo" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Rtl" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Size" nillable="true" type="tns:Point"/>
    <xs:element minOccurs="0" name="Slide" nillable="true" type="tns:EditSlide"/>
    <xs:element minOccurs="0" name="SlideOrder" nillable="true" type="tns:SlideIdList"/>
    <xs:element minOccurs="0" name="Undo" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

CompatProblems: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that SHOULD **<7>** specify whether compatibility issues may occur if the presentation is edited. This element MUST be present.

EditSlideInfos: An [ArrayOfEditSlideInfoFe](#) that specifies information about presentation slides. This element MUST be present.

FCreateNew: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation is new.

FirstSlideIndex: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the index of the first presentation slide. This element MUST be present.

HasComments: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation contains comments.

MainMasters: An [ArrayOfMainMasterFe](#) elements that specify information about main master slides. This element MUST be present.

Redo: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether [Redo](#) is possible. This element MUST be present.

Rtl: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation content is **right-to-left**. This element MUST be present.

Size: A [Point](#) that specifies the dimensions of the presentation slides. This element MUST be present.

Slide: Reserved. MUST be ignored.

SlideOrder: A [SlideIdList](#) that specifies the order of the presentation slides. This element MUST be present.

Undo: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether [Undo](#) is possible. This element MUST be present.

2.2.4.14 EditShape

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a shape.

```
<xs:complexType name="EditShape">
  <xs:sequence>
    <xs:element minOccurs="0" name="Alt" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="EditBounds" nillable="true" type="tns:Rectangle"/>
    <xs:element minOccurs="0" name="EditHtml" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="FontSizeMapping" nillable="true"
type="tns:FontSizeMapping"/>
    <xs:element minOccurs="0" name="IsPicture" type="xs:boolean"/>
    <xs:element minOccurs="0" name="IsPlaceholder" type="xs:boolean"/>
    <xs:element minOccurs="0" name="IsPrompt" type="xs:boolean"/>
    <xs:element minOccurs="0" name="IsSmartArt" type="xs:boolean"/>
    <xs:element minOccurs="0" name="PlaceholderType" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="ShapeId" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="ShapeViewElemId" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="SmartArtInfo" nillable="true" type="tns:SmartArtInfo"/>
    <xs:element minOccurs="0" name="TextViewElemId" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>
```

Alt: Reserved. MUST be ignored.

EditBounds: A [Rectangle](#) that specifies the bounds of the shape. This element MUST be present.

EditHtml: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the text content of the shape as HTML. This element MUST be present.

FontSizeMapping: Reserved. MUST be ignored.

IsPicture: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the shape is a picture. This element MUST be present.

IsPlaceholder: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the shape is a **placeholder**. This element MUST be present.

IsPrompt: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the shape contains text. This element MUST be present.

IsSmartArt: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the shape is a SmartArt diagram. This element MUST be present.

PlaceholderType: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the type of placeholder. This element MUST be present.

ShapeId: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the identifier of the shape. This element MUST be present.

ShapeViewElemId: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the identifier of the [ViewElement](#) for the shape. This element MUST be present.

SmartArtInfo: A [SmartArtInfo](#) element that specifies information about the SmartArt diagram. MUST be present if and only if **IsSmartArt** is true.

TextViewElemId: An **xs:unsignedInt** ([XMLSCHEMA2](#) section 3.3.22) that specifies the identifier of the ViewElement for the text content of the shape.

2.2.4.15 EditSlide

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a presentation slide.

```
<xs:complexType name="EditSlide">
  <xs:sequence>
    <xs:element minOccurs="0" name="ContentMasterId" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="EditShapes" nillable="true" type="tns:ArrayOfEditShape"/>
    <xs:element minOccurs="0" name="Hidden" type="xs:boolean"/>
    <xs:element minOccurs="0" name="IsTitleContentMaster" type="xs:boolean"/>
    <xs:element minOccurs="0" name="MainMasterId" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Notes" nillable="true" type="tns:Notes"/>
    <xs:element minOccurs="0" name="SlideId" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Theme" nillable="true" type="tns:Theme"/>
    <xs:element minOccurs="0" name="ThumbnailViewElement" nillable="true"
type="tns:ViewElement"/>
    <xs:element minOccurs="0" name="ViewElems" nillable="true"
type="tns:ArrayOfViewElement"/>
  </xs:sequence>
</xs:complexType>
```

ContentMasterId: An **xs:unsignedInt** ([XMLSCHEMA2](#) section 3.3.22) that specifies the identifier of the [ContentMasterFe](#) for the presentation slide. This element MUST be present.

EditShapes: An [ArrayOfEditShape](#) element that specifies the shapes of the presentation slide. This element MUST be present.

Hidden: An **xs:boolean** ([XMLSCHEMA2](#) section 3.2.2) that specifies whether the presentation slide is hidden. This element MUST be present.

IsTitleContentMaster: An **xs:boolean** ([XMLSCHEMA2](#) section 3.2.2) that specifies whether the slide layout specified by **ContentMasterId** is a **title master slide**. This element MUST be present.

MainMasterId: An **xs:unsignedInt** ([XMLSCHEMA2](#) section 3.3.22) that specifies the identifier of the main master slide for this presentation slide. This element MUST be present.

Notes: A [Notes](#) element that specifies the presentation notes for the presentation slide. This element MUST be present.

SlideId: An **xs:unsignedInt** ([XMLSCHEMA2](#) section 3.3.22) that specifies the identifier of the presentation slide. This element MUST be present.

Theme: A [Theme](#) element that specifies the theme for the presentation slide. MUST be present.

ThumbnailViewElement: A [ViewElement](#) element that specifies the thumbnail of the presentation slide. This element MUST be present.

ViewElems: An [ArrayOfViewElement](#) element that specifies the ViewElement elements for slide content of the presentation slide. This element MUST be present.

2.2.4.16 EditSlideChanges

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies the content of the presentation slide that is modified as a result of a request to the protocol server.

```
<xs:complexType name="EditSlideChanges">
  <xs:sequence>
    <xs:element xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      minOccurs="0" name="DelElems" nillable="true" type="q2:ArrayOfunsignedInt"/>
    <xs:element minOccurs="0" name="Elems" nillable="true" type="tns:ArrayOfViewElement"/>
    <xs:element minOccurs="0" name="Hidden" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Id" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="ModifiedShapes" nillable="true"
      type="tns:ArrayOfEditShape"/>
    <xs:element minOccurs="0" name="Notes" nillable="true" type="tns:Notes"/>
    <xs:element minOccurs="0" name="Refresh" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Thumb" nillable="true" type="tns:ViewElement"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

DelElems: Reserved. MUST be ignored.

Elems: An [ArrayOfViewElement](#) element that specifies the [ViewElement](#) elements for slide content on the presentation slide.

Hidden: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation slide is hidden.

Id: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the identifier of the presentation slide. This element MUST be present.

ModifiedShapes: An [ArrayOfEditShape](#) element that specifies the shapes modified.

Notes: A [Notes](#) element that specifies the presentation notes for the presentation slide.

Refresh: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation slide has outdated ViewElement elements.

Thumb: A ViewElement element that specifies the thumbnail of the presentation slide.

Title: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the title of the presentation slide.

2.2.4.17 EditSlideInfoFe

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies information about a presentation slide.

```
<xs:complexType name="EditSlideInfoFe">
  <xs:sequence>
    <xs:element minOccurs="0" name="Hidden" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Id" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Thumb" nillable="true" type="tns:ViewElement"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```

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

```

Hidden: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation slide is hidden.

Id: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies an identifier for the presentation slide.

Thumb: A [ViewElement](#) that specifies the thumbnail of the presentation slide.

Title: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the title of the presentation slide.

2.2.4.18 FontSizeMapping

Namespace: <http://schemas.datacontract.org/2004/07/p>

Reserved. MUST be ignored.

```

<xs:complexType name="FontSizeMapping">
  <xs:sequence>
    <xs:element minOccurs="0" name="Size1" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size2" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size3" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size4" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size5" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size6" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="Size7" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>

```

Size1: Reserved. MUST be ignored.

Size2: Reserved. MUST be ignored.

Size3: Reserved. MUST be ignored.

Size4: Reserved. MUST be ignored.

Size5: Reserved. MUST be ignored.

Size6: Reserved. MUST be ignored.

Size7: Reserved. MUST be ignored.

2.2.4.19 Html

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies HTML content.

```

<xs:complexType name="Html">
  <xs:sequence>
    <xs:element minOccurs="0" name="Text" nillable="true" type="xs:string"/>
  </xs:sequence>

```

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

Text: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the HTML content.

2.2.4.20 MainMasterFe

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a main master slide.

```
<xs:complexType name="MainMasterFe">
  <xs:sequence>
    <xs:element minOccurs="0" name="ContentMasters" nillable="true"
type="tns:ArrayOfContentMasterFe"/>
    <xs:element minOccurs="0" name="Id" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="InContent" type="xs:boolean"/>
    <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

ContentMasters: An [ArrayOfContentMasterFe](#) element that specifies slide layouts.

Id: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the identifier of the main master slide.

InContent: Reserved. MUST be ignored.

Name: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the name of the main master slide.

2.2.4.21 Notes

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies the notes of a presentation slide.

```
<xs:complexType name="Notes">
  <xs:sequence>
    <xs:element minOccurs="0" name="Html" nillable="true" type="tns:Html"/>
    <xs:element minOccurs="0" name="IsPrompt" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>
```

Html: An [Html](#) element that specifies the content of the notes.

IsPrompt: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the notes is empty.

2.2.4.22 Point

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a two dimensional point.

```

<xs:complexType name="Point">
  <xs:sequence>
    <xs:element minOccurs="0" name="X" type="xs:int"/>
    <xs:element minOccurs="0" name="Y" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

X: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the x coordinate. This element **MUST** be present.

Y: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the y coordinate. This element **MUST** be present.

2.2.4.23 PptViewingService.PrintResult

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that **SHOULD** [<8>](#) specify the result of a print operation.

```

<xs:complexType name="PptViewingService.PrintResult">
  <xs:sequence>
    <xs:element minOccurs="0" name="PrintUrl" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

```

PrintUrl: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) element that specifies the URL for getting a printable version of the presentation.

2.2.4.24 Rectangle

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a rectangle.

```

<xs:complexType name="Rectangle">
  <xs:sequence>
    <xs:element minOccurs="0" name="Bottom" type="xs:int"/>
    <xs:element minOccurs="0" name="Left" type="xs:int"/>
    <xs:element minOccurs="0" name="Right" type="xs:int"/>
    <xs:element minOccurs="0" name="Top" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

Bottom: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the location of the bottom of the rectangle. This element **MUST** be present.

Left: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the location of the left side of the rectangle. This element **MUST** be present.

Right: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the location of the right side of the rectangle. This element **MUST** be present.

Top: An **xs:int** ([\[XMLSCHEMA2\]](#) section 3.3.17) that specifies the location of the top of the rectangle. This element **MUST** be present.

2.2.4.25 ServiceError

Namespace: http://schemas.datacontract.org/2004/07/p

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

```
<xs:complexType name="ServiceError">
  <xs:sequence>
    <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string"/>
    <xs:element
      xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Powerpoint.Pipe.Interface"
      minOccurs="0" name="RecommendedActions" type="q3:ClientActions"/>
    <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Type" type="tns:ServiceErrorType"/>
  </xs:sequence>
</xs:complexType>
```

Message: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) that specifies the error message description. MUST be present. The string length MUST be greater than zero if the **Type** element has a value of [ApplicationError](#).

RecommendedActions: Reserved. MUST be ignored.

Title: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) that specifies the error title. MUST be present. The string length MUST be greater than zero if the **Type** element has a value of [ApplicationError](#).

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

2.2.4.26 ServiceResult

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies the result of a protocol method. The protocol server returns this type to the protocol client containing either a successful **Result** element or an **Error** element.

```
<xs:complexType name="ServiceResult">
  <xs:sequence>
    <xs:element minOccurs="0" name="Error" nillable="true" type="tns:ServiceError"/>
    <xs:element minOccurs="0" name="Result" nillable="true" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
```

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

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

2.2.4.27 SlideIdList

Namespace: http://schemas.datacontract.org/2004/07/p

A complex type that specifies a list of identifiers for presentation slides.

```

<xs:complexType name="SlideIdList">
  <xs:sequence>
    <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      minOccurs="0" name="SlideIds" nillable="true" type="q1:ArrayOfunsignedInt"/>
  </xs:sequence>
</xs:complexType>

```

SlideIds: An [ArrayOfunsignedInt](#) element that specifies the list of identifiers.

2.2.4.28 SmartArtInfo

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies information about a SmartArt diagram.

```

<xs:complexType name="SmartArtInfo">
  <xs:sequence>
    <xs:element minOccurs="0" name="fReverseChecked" type="xs:boolean"/>
    <xs:element minOccurs="0" name="fReverseEnabled" type="xs:boolean"/>
  </xs:sequence>
</xs:complexType>

```

fReverseChecked: An **xs:boolean** ([XMLSCHEMA2](#) section 3.2.2) that specifies whether the SmartArt diagram is reversed.

fReverseEnabled: An **xs:boolean** ([XMLSCHEMA2](#) section 3.2.2) that specifies whether the SmartArt diagram can be reversed.

2.2.4.29 Theme

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies information about a theme.

```

<xs:complexType name="Theme">
  <xs:sequence>
    <xs:element minOccurs="0" name="Accent1" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Accent2" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Accent3" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Accent4" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Accent5" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Accent6" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Dark1" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Dark2" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Light1" nillable="true" type="tns:ThemeColor"/>
    <xs:element minOccurs="0" name="Light2" nillable="true" type="tns:ThemeColor"/>
  </xs:sequence>
</xs:complexType>

```

Accent1: A [ThemeColor](#) element that defines the first accent color in the theme.

Accent2: A ThemeColor element that defines the second accent color in the theme.

Accent3: A ThemeColor element that defines the third accent color in the theme.

Accent4: A ThemeColor element that defines the fourth accent color in the theme.

Accent5: A ThemeColor element that defines the fifth accent color in the theme.

Accent6: A ThemeColor element that defines the sixth accent color in the theme.

Dark1: A ThemeColor element that defines a dark color in the theme.

Dark2: A ThemeColor element that defines a second dark color in the theme.

Light1: A ThemeColor element that defines a light color in the theme.

Light2: A ThemeColor element that defines a second light color in the theme.

2.2.4.30 ThemeColor

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a color using the red, green, blue RGB color model. Red, green, and blue are each expressed as an unsigned byte from 0 for no intensity to 255 for full intensity.

```
<xs:complexType name="ThemeColor">
  <xs:sequence>
    <xs:element minOccurs="0" name="b" type="xs:unsignedByte"/>
    <xs:element minOccurs="0" name="g" type="xs:unsignedByte"/>
    <xs:element minOccurs="0" name="r" type="xs:unsignedByte"/>
  </xs:sequence>
</xs:complexType>
```

b: An **xs:unsignedByte** ([\[XMLSCHEMA2\]](#) section 3.3.24) that specifies the blue component of the color. This element **MUST** be present.

g: An **xs:unsignedByte** ([\[XMLSCHEMA2\]](#) section 3.3.24) that specifies the green component of the color. This element **MUST** be present.

r: An **xs:unsignedByte** ([\[XMLSCHEMA2\]](#) section 3.3.24) that specifies the red component of the color. This element **MUST** be present.

2.2.4.31 ViewElement

Namespace: <http://schemas.datacontract.org/2004/07/p>

A complex type that specifies a visual representation of presentation slide content.

```
<xs:complexType name="ViewElement">
  <xs:sequence>
    <xs:element minOccurs="0" name="Alt" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Clip" nillable="true" type="tns:ClippingInfo"/>
    <xs:element minOccurs="0" name="Html" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Id" type="xs:unsignedInt"/>
    <xs:element minOccurs="0" name="ImgId" nillable="true" type="xs:string"/>
    <xs:element minOccurs="0" name="Position" nillable="true" type="tns:Point"/>
    <xs:element minOccurs="0" name="Z" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>
```


Alt: Reserved. MUST be ignored.

Clip: A [ClippingInfo](#) that specifies the region within the image. This element MUST be present.

Html: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies text content. MUST be present if and only if **ImgId** is not present.

Id: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the identifier. This element MUST be present.

ImgId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the URL of the image.[<9>](#)

Position: A [Point](#) element that specifies the position.

Z: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the z-order.

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
ClientActions	Reserved. MUST be ignored.
ServiceErrorType	This simple type specifies an enumeration of a set of protocol errors returned by the protocol server to the protocol client.
char	Reserved. MUST be ignored. 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. MUST be ignored. 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	Reserved. MUST be ignored. 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.

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. MUST be ignored. 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.
ClientActions	Reserved. MUST be ignored.
duration	Reserved. MUST be ignored. 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	Reserved. MUST be ignored. The client and the server MUST NOT use this as the type

Simple Type	Description
	of an element, and the client and the server MUST ignore it, if receiving an element of this type.
ServiceErrorType	A simple type that specifies an enumeration of a set of protocol errors returned by the protocol server to the protocol client.

2.2.5.1 char

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

Reserved. MUST be ignored. 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">
  <xs:restriction base="xs:int"/>
</xs:simpleType>
```

2.2.5.2 ClientActions

Namespace:

http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Powerpoint.Pipe.Interface

Reserved. MUST be ignored.

```
<xs:simpleType name="ClientActions">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Dismiss"/>
        <xs:enumeration value="Close"/>
        <xs:enumeration value="OpenInClient"/>
        <xs:enumeration value="Refresh"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
```

The following table specifies the allowable values for ClientActions:

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

2.2.5.3 duration

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

Reserved. MUST be ignored. 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">
  <xs:restriction base="xs:duration">
    <xs:pattern value="\-?P(\d*D)?(T(\d*M)?(\d*(\.\d*)?S)?)?" />
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S" />
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S" />
  </xs:restriction>
</xs:simpleType>
```

2.2.5.4 guid

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

Reserved. MUST be ignored. 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="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>
```

2.2.5.5 ServiceErrorType

Namespace: http://schemas.datacontract.org/2004/07/p

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

```
<xs:simpleType name="ServiceErrorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UnknownError" />
    <xs:enumeration value="ApplicationError" />
    <xs:enumeration value="Timeout" />
    <xs:enumeration value="ServiceBusy" />
    <xs:enumeration value="SessionFull" />
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ServiceErrorType:

Value	Meaning
UnknownError	The protocol server encountered an unknown error.
ApplicationError	The protocol server encountered an application error.

Value	Meaning
Timeout	The protocol server timed out.
ServiceBusy	Reserved. MUST be ignored.
SessionFull	Reserved. MUST be ignored.

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**, **present**, and **not null**.

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

3.1 Server Details

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



Figure 2: Sample communication between protocol client and protocol server

The protocol client sends the **IPptService_GetPresentationId_InputMessage SOAP message** as specified in section [3.1.4.11](#) to obtain a valid identifier for a presentation within an **IPptService_GetPresentationId_OutputMessage**.

The protocol client sends an **IPptService_GetEditPresInfo_InputMessage** message and the protocol server responds with an **IPptService_GetEditPresInfo_OutputMessage** containing information about the requested presentation.

The protocol client sends one or more **IPptService_GetEditSlide_InputMessage** messages and the protocol server responds with an **IPptService_GetSlide_OutputMessage** containing information about the requested presentation slide.

The protocol client sends one or more **IPptService_ReplaceText_InputMessage** messages and the protocol server responds with an **IPptService_ReplaceText_OutputMessage** containing information about the result of the command.

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.

3.1.2 Timers

None.

3.1.3 Initialization

The protocol server MUST expose its Web methods at the following URL, which builds upon a base URL. The URL MUST conform to the following structure: *base URL/_vti_bin/ppt/edit.svc*. 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 message processing events and sequencing rules are listed in this section. [<10>](#)

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

WSDL Operation	Description
ApplyTheme	The ApplyTheme operation SHOULD <11> be used to change the theme applied to a presentation.
ChangePictureStyle	The ChangePictureStyle operation is used to change the style applied to a picture.
ChangeSmartArtColor	The ChangeSmartArtColor operation is used to change the color transform applied to a SmartArt diagram.
ChangeSmartArtLayout	The ChangeSmartArtLayout operation is used to change the layout applied to a SmartArt diagram.
ChangeSmartArtStyle	The ChangeSmartArtStyle operation is used to change the style applied to a SmartArt diagram.
ClearPlaceholder	The ClearPlaceholder is used to delete a shape in a presentation.
DeleteSlide	The DeleteSlide operation is used to delete a presentation slide from a presentation.
DuplicateSlide	The DuplicateSlide operation is used to duplicate a presentation slide.
GetEditPresInfo	The GetEditPresInfo operation is used to retrieve information about a presentation.
GetEditSlide	The GetEditSlide operation is used to retrieve information about a presentation slide.

WSDL Operation	Description
GetPresentationId	The GetPresentationId operation is used to retrieve an identifier for a presentation.
InsertClipart	The InsertClipart operation SHOULD <12> be used to insert a picture into a presentation.
InsertSlide	The InsertSlide operation is used to insert a presentation slide into a presentation.
InsertSmartArt	The InsertSmartArt operation is used to insert a SmartArt diagram into a presentation.
MoveSlide	The MoveSlide operation is used to move a presentation slide in a presentation.
Noop	Reserved. MUST not be called.
Print	A Print operation SHOULD <13> be used to print a presentation.
Redo	The Redo operation is used to redo the last command in a presentation that is undone.
ReplaceNotes	The ReplaceNotes operation is used to replace presentation notes in a presentation.
ReplaceText	The ReplaceText operation is used to replace text in a shape in a presentation.
ResetSmartArt	The ResetSmartArt operation is used to remove all customizations on a SmartArt diagram in a presentation.
ReverseSmartArt	The ReverseSmartArt operation is used to reverse the direction of a SmartArt diagram in a presentation.
SaveAndClose	The SaveAndClose operation is used to save the contents of a presentation and unlock it.
ShowHideSlide	The ShowHideSlide operation is used to show or hide a presentation slide.
Undo	The Undo operation is used to undo the last command in a presentation that is not undone.
UpdateCanary	Reserved. MUST be ignored.
WaitCrash	Reserved. MUST not be called.

3.1.4.1 ApplyTheme

The **ApplyTheme** operation SHOULD [<14>](#) be used to change the theme applied to a presentation.

```

<wsdl:operation name="ApplyTheme">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ApplyTheme"
    message="tns:IPptEditInternal_ApplyTheme_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ApplyThemeResponse"
  </wsdl:output>
</wsdl:operation>

```

```

message="tns:IPptEditInternal_ApplyTheme_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

3.1.4.1.1 Messages

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

3.1.4.1.1.1 IPptEditInternal_ApplyTheme_InputMessage

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

The **SOAP action** value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ApplyTheme

```

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

3.1.4.1.1.2 IPptEditInternal_ApplyTheme_OutputMessage

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

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

3.1.4.1.2 Elements

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

3.1.4.1.2.1 ApplyTheme

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

```

<xs:element name="ApplyTheme">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="themeName" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide.

themeName: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the name of the theme to apply.

3.1.4.1.2.2 ApplyThemeResponse

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

```
<xs:element name="ApplyThemeResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q4="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ApplyThemeResult" nillable="true" type="q4:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ApplyThemeResult: A **ServiceResult** (section [2.2.4.26](#)) that specifies the result of the operation. The **Result** child element MUST be an **EditCommandResponse** (section [2.2.4.11](#)).

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 ChangePictureStyle

The **ChangePictureStyle** operation is used to change the style applied to a picture.

```
<wsdl:operation name="ChangePictureStyle">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangePictureStyle"
message="tns:IPptEditInternal_ChangePictureStyle_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangePictureStyleResponse"
message="tns:IPptEditInternal_ChangePictureStyle_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_ChangePictureStyle_InputMessage** request message, and the protocol server responds with an **IPptService_ChangePictureStyle_OutputMessage** response message.

3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.2.1.1 IPptEditInternal_ChangePictureStyle_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ChangePictureStyle
```

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

3.1.4.2.1.2 IPptEditInternal_ChangePictureStyle_OutputMessage

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

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

3.1.4.2.2 Elements

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

3.1.4.2.2.1 ChangePictureStyle

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

```
<xs:element name="ChangePictureStyle">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="pictureStyleId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the picture.

pictureStyleId: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies which style the picture is changed to. The value MUST be in the range from 15505 through 15532.

3.1.4.2.2.2 ChangePictureStyleResponse

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

```
<xs:element name="ChangePictureStyleResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q18="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangePictureStyleResult" nillable="true" type="q18:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ChangePictureStyleResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 ChangeSmartArtColor

The **ChangeSmartArtColor** operation is used to change the color transform applied to a SmartArt diagram.

```
<wsdl:operation name="ChangeSmartArtColor">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtColor"
message="tns:IPptEditInternal_ChangeSmartArtColor_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtColorResponse"
message="tns:IPptEditInternal_ChangeSmartArtColor_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_ChangeSmartArtColor_InputMessage** request message, and the protocol server responds with an **IPptService_ChangeSmartArtColor_OutputMessage** response message.

3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.3.1.1 IPptEditInternal_ChangeSmartArtColor_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ChangeSmartArtColor
```

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

3.1.4.3.1.2 IPptEditInternal_ChangeSmartArtColor_OutputMessage

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

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

3.1.4.3.2 Elements

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

3.1.4.3.2.1 ChangeSmartArtColor

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

```
<xs:element name="ChangeSmartArtColor">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="colorTransUri" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation where the color transform of the SmartArt diagram is changed, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide where the color transform of the SmartArt diagram is changed.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the SmartArt diagram the color of which is changed.

colorTransUri: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the color transform which is applied to the SmartArt diagram.

category: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the category to which the color transform belongs.

3.1.4.3.2.2 ChangeSmartArtColorResponse

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

```
<xs:element name="ChangeSmartArtColorResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q23="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtColorResult" nillable="true" type="q23:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ChangeSmartArtColorResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

None.

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 ChangeSmartArtLayout

The **ChangeSmartArtLayout** operation is used to change the layout applied to a SmartArt diagram.

```
<wsdl:operation name="ChangeSmartArtLayout">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtLayout"
message="tns:IPptEditInternal_ChangeSmartArtLayout_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
```

```

ServerInternalService/IPptEdit/ChangeSmartArtLayoutResponse"
message="tns:IPptEditInternal_ChangeSmartArtLayout_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ChangeSmartArtLayout_InputMessage** request message, and the protocol server responds with an **IPptService_ChangeSmartArtLayout_OutputMessage** response message.

3.1.4.4.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.4.1.1 IPptEditInternal_ChangeSmartArtLayout_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ChangeSmartArtLayout

```

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

3.1.4.4.1.2 IPptEditInternal_ChangeSmartArtLayout_OutputMessage

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

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

3.1.4.4.2 Elements

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

3.1.4.4.2.1 ChangeSmartArtLayout

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

```

<xs:element name="ChangeSmartArtLayout">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="layoutUri" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation where the layout of the SmartArt diagram is changed, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide where the layout of the SmartArt diagram is changed.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the SmartArt diagram the layout of which is changed.

layoutUri: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the layout which is applied to the SmartArt diagram.

category: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the category to which the layout belongs.

3.1.4.4.2.2 ChangeSmartArtLayoutResponse

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

```
<xs:element name="ChangeSmartArtLayoutResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q22="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtLayoutResult" nillable="true" type="q22:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ChangeSmartArtLayoutResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 ChangeSmartArtStyle

The **ChangeSmartArtStyle** operation is used to change the style applied to a SmartArt diagram.

```
<wsdl:operation name="ChangeSmartArtStyle">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
```

```

ServerInternalService/IPptEdit/ChangeSmartArtStyle"
message="tns:IPptEditInternal_ChangeSmartArtStyle_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtStyleResponse"
message="tns:IPptEditInternal_ChangeSmartArtStyle_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ChangeSmartArtStyle_InputMessage** request message, and the protocol server responds with an **IPptService_ChangeSmartArtStyle_OutputMessage** response message.

3.1.4.5.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.5.1.1 IPptEditInternal_ChangeSmartArtStyle_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ChangeSmartArtStyle

```

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

3.1.4.5.1.2 IPptEditInternal_ChangeSmartArtStyle_OutputMessage

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

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

3.1.4.5.2 Elements

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

3.1.4.5.2.1 ChangeSmartArtStyle

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

```

<xs:element name="ChangeSmartArtStyle">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="styleUri" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```


presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation where the style of the SmartArt diagram is changed, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide where the style of the SmartArt diagram is changed.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the SmartArt diagram the style of which is changed.

styleUri: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the style which is applied to the SmartArt diagram.

category: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the category to which the style belongs.

3.1.4.5.2.2 ChangeSmartArtStyleResponse

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

```
<xs:element name="ChangeSmartArtStyleResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q24="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtStyleResult" nillable="true" type="q24:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ChangeSmartArtStyleResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 ClearPlaceholder

The **ClearPlaceholder** is used to delete a shape in a presentation.

```

<wsdl:operation name="ClearPlaceholder">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ClearPlaceholder"
    message="tns:IPptEditInternal_ClearPlaceholder_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ClearPlaceholderResponse"
    message="tns:IPptEditInternal_ClearPlaceholder_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ClearPlaceholder_InputMessage** request message, and the protocol server responds with an **IPptService_ClearPlaceholder_OutputMessage** response message.

3.1.4.6.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.6.1.1 IPptEditInternal_ClearPlaceholder_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ClearPlaceholder

```

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

3.1.4.6.1.2 IPptEditInternal_ClearPlaceholder_OutputMessage

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

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

3.1.4.6.2 Elements

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

3.1.4.6.2.1 ClearPlaceholder

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

```

<xs:element name="ClearPlaceholder">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>

```

```
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the shape that is deleted, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide containing the shape that is deleted.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the shape that is deleted.

3.1.4.6.2.2 ClearPlaceholderResponse

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

```
<xs:element name="ClearPlaceholderResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ClearPlaceholderResult" nillable="true" type="q1:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ClearPlaceholderResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 DeleteSlide

The **DeleteSlide** operation is used to delete a presentation slide from a presentation.

```
<wsdl:operation name="DeleteSlide">
```

```

    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DeleteSlide"
message="tns:IPptEditInternal_DeleteSlide_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DeleteSlideResponse"
message="tns:IPptEditInternal_DeleteSlide_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_DeleteSlide_InputMessage** request message, and the protocol server responds with an **IPptService_DeleteSlide_OutputMessage** response message.

3.1.4.7.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.7.1.1 IPptEditInternal_DeleteSlide_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/DeleteSlide

```

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

3.1.4.7.1.2 IPptEditInternal_DeleteSlide_OutputMessage

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

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

3.1.4.7.2 Elements

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

3.1.4.7.2.1 DeleteSlide

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

```

<xs:element name="DeleteSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide to delete.

3.1.4.7.2.2 DeleteSlideResponse

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

```
<xs:element name="DeleteSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q12="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="DeleteSlideResult" nillable="true" type="q12:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

DeleteSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.7.3 Complex Types

None.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 DuplicateSlide

The **DuplicateSlide** operation is used to duplicate a presentation slide.

```
<wsdl:operation name="DuplicateSlide">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DuplicateSlide"
message="tns:IPptEditInternal_DuplicateSlide_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DuplicateSlideResponse"
```

```

message="tns:IPptEditInternal_DuplicateSlide_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_DuplicateSlide_InputMessage** request message, and the protocol server responds with an **IPptService_DuplicateSlide_OutputMessage** response message.

3.1.4.8.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.8.1.1 IPptEditInternal_DuplicateSlide_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/DuplicateSlide

```

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

3.1.4.8.1.2 IPptEditInternal_DuplicateSlide_OutputMessage

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

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

3.1.4.8.2 Elements

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

3.1.4.8.2.1 DuplicateSlide

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

```

<xs:element name="DuplicateSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="srcId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

srcId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide to duplicate.

dstId: A **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide that precedes the location of the duplicated slide.

3.1.4.8.2.2 DuplicateSlideResponse

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

```
<xs:element name="DuplicateSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q13="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="DuplicateSlideResult" nillable="true" type="q13:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

DuplicateSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.8.3 Complex Types

None.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 GetEditPresInfo

The **GetEditPresInfo** operation is used to retrieve information about a presentation.

```
<wsdl:operation name="GetEditPresInfo">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditPresInfo"
message="tns:IPptEditInternal_GetEditPresInfo_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditPresInfoResponse"
message="tns:IPptEditInternal_GetEditPresInfo_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_GetEditPresInfo_InputMessage** request message, and the protocol server responds with an **IPptService_GetEditPresInfo_OutputMessage** response message.

3.1.4.9.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.9.1.1 IPptEditInternal_GetEditPresInfo_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/GetEditPresInfo
```

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

3.1.4.9.1.2 IPptEditInternal_GetEditPresInfo_OutputMessage

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

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

3.1.4.9.2 Elements

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

3.1.4.9.2.1 GetEditPresInfo

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

```
<xs:element name="GetEditPresInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="firstSlideIndex" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="numSlides" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="includedEditSlideIndex" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

firstSlideIndex: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the index of the first presentation slide in the slides of the presentation.

numSlides: An **xs:unsignedInt** ([\[XMLSCHEMA2\]](#) section 3.3.22) that specifies the count of presentation slides for which to request information.

includedEditSlideIndex: Reserved. MUST be ignored.

3.1.4.9.2.2 GetEditPresInfoResponse

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

```
<xs:element name="GetEditPresInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q6="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="GetEditPresInfoResult" nillable="true" type="q6:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetEditPresInfoResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditPresentationInfo](#).

3.1.4.9.3 Complex Types

None.

3.1.4.9.4 Simple Types

None.

3.1.4.9.5 Attributes

None.

3.1.4.9.6 Groups

None.

3.1.4.9.7 Attribute Groups

None.

3.1.4.10 GetEditSlide

The **GetEditSlide** operation is used to retrieve information about a presentation slide.

```
<wsdl:operation name="GetEditSlide">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditSlide"
message="tns:IPptEditInternal_GetEditSlide_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditSlideResponse"
message="tns:IPptEditInternal_GetEditSlide_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_GetEditSlide_InputMessage** request message, and the protocol server responds with an **IPptService_GetEditSlide_OutputMessage** response message.

3.1.4.10.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.10.1.1 IPptEditInternal_GetEditSlide_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/GetEditSlide
```

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

3.1.4.10.1.2 IPptEditInternal_GetEditSlide_OutputMessage

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

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

3.1.4.10.2 Elements

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

3.1.4.10.2.1 GetEditSlide

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

```
<xs:element name="GetEditSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide.

3.1.4.10.2.2 GetEditSlideResponse

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

```
<xs:element name="GetEditSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q7="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="GetEditSlideResult" nillable="true" type="q7:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

GetEditSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditSlide](#).

3.1.4.10.3 Complex Types

None.

3.1.4.10.4 Simple Types

None.

3.1.4.10.5 Attributes

None.

3.1.4.10.6 Groups

None.

3.1.4.10.7 Attribute Groups

None.

3.1.4.11 GetPresentationId

The **GetPresentationId** operation is used to retrieve an identifier for a presentation.

```
<wsdl:operation name="GetPresentationId">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/GetPresentationId"
    message="tns:IPptEditInternal_GetPresentationId_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/GetPresentationIdResponse"
    message="tns:IPptEditInternal_GetPresentationId_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_GetPresentationId_InputMessage** request message, and the protocol server responds with an **IPptService_GetPresentationId_OutputMessage** response message.

3.1.4.11.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.11.1.1 IPptEditInternal_GetPresentationId_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/GetPresentationId
```

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

3.1.4.11.1.2 IPptEditInternal_GetPresentationId_OutputMessage

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

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

3.1.4.11.2 Elements

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

3.1.4.11.2.1 GetPresentationId

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

```
<xs:element name="GetPresentationId">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="rawDocId" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

rawDocId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the **absolute URL** to the presentation.

3.1.4.11.2.2 GetPresentationIdResponse

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

```
<xs:element name="GetPresentationIdResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="GetPresentationIdResult" nillable="true"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetPresentationIdResult: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation.

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 InsertClipart

The **InsertClipart** operation SHOULD [<15>](#) be used to insert a picture into a presentation.

```
<wsdl:operation name="InsertClipart">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/InsertClipart"
    message="tns:IPptEditInternal_InsertClipart_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/InsertClipartResponse"
    message="tns:IPptEditInternal_InsertClipart_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_InsertClipart_InputMessage** request message, and the protocol server responds with an **IPptService_InsertClipart_OutputMessage** response message.

3.1.4.12.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.12.1.1 IPptEditInternal_InsertClipart_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/InsertClipart
```

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

3.1.4.12.1.2 IPptEditInternal_InsertClipart_OutputMessage

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

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

3.1.4.12.2 Elements

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

3.1.4.12.2.1 InsertClipart

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

```
<xs:element name="InsertClipart">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="assetId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="Lcid" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="alt" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation into which the picture is inserted, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide into which the picture is inserted.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the placeholder shape into which the picture is inserted.

assetId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the unique identifier of the picture.

Lcid: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the **language code identifier (LCID)**, as specified in [\[MS-LCID\]](#), of the picture.

alt: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the alternate text for the picture.

3.1.4.12.2.2 InsertClipartResponse

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

```
<xs:element name="InsertClipartResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q5="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="InsertClipartResult" nillable="true" type="q5:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

InsertClipartResult: A **ServiceResult** (section [2.2.4.26](#)) that specifies the result of the operation. The **Result** child element MUST be an **EditCommandResponse** (section [2.2.4.11](#)).

3.1.4.12.3 Complex Types

None.

3.1.4.12.4 Simple Types

None.

3.1.4.12.5 Attributes

None.

3.1.4.12.6 Groups

None.

3.1.4.12.7 Attribute Groups

None.

3.1.4.13 InsertSlide

The **InsertSlide** operation is used to insert a presentation slide into a presentation.

```
<wsdl:operation name="InsertSlide">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/InsertSlide"
    message="tns:IPptEditInternal_InsertSlide_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    <wsdl:output
      wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
      ServerInternalService/IPptEdit/InsertSlideResponse"
      message="tns:IPptEditInternal_InsertSlide_OutputMessage"
      xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:output>
  </wsdl:operation>
```

The protocol client sends an **IPptService_InsertSlide_InputMessage** request message, and the protocol server responds with an **IPptService_InsertSlide_OutputMessage** response message.

3.1.4.13.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.13.1.1 IPptEditInternal_InsertSlide_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/InsertSlide
```

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

3.1.4.13.1.2 IPptEditInternal_InsertSlide_OutputMessage

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

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

3.1.4.13.2 Elements

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

3.1.4.13.2.1 InsertSlide

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

```
<xs:element name="InsertSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="mainMasterId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="contentMasterId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

dstId: A **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide that precedes the location of the inserted slide. A value of zero indicates that no presentation slide precedes the location. In this case, the location of the inserted presentation slide will be the first slide of the presentation.

mainMasterId: An **ST_SlideMasterId** ([\[ISO/IEC-29500-1\]](#) section 19.7.16) that specifies the identifier for the main master slide of the inserted presentation slide.

contentMasterId: An **ST_SlideLayoutId** ([\[ISO/IEC-29500-1\]](#) section 19.7.14) that specifies the identifier for the slide layout of the inserted presentation slide.

3.1.4.13.2.2 InsertSlideResponse

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

```
<xs:element name="InsertSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q9="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="InsertSlideResult" nillable="true" type="q9:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

InsertSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.13.3 Complex Types

None.

3.1.4.13.4 Simple Types

None.

3.1.4.13.5 Attributes

None.

3.1.4.13.6 Groups

None.

3.1.4.13.7 Attribute Groups

None.

3.1.4.14 InsertSmartArt

The **InsertSmartArt** operation is used to insert a SmartArt diagram into a presentation

```
<wsdl:operation name="InsertSmartArt">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/InsertSmartArt"
    message="tns:IPptEditInternal_InsertSmartArt_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    <wsdl:output
      wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
      ServerInternalService/IPptEdit/InsertSmartArtResponse"
      message="tns:IPptEditInternal_InsertSmartArt_OutputMessage"
      xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:output>
  </wsdl:operation>
```

The protocol client sends an **IPptService_InsertSmartArt_InputMessage** request message, and the protocol server responds with an **IPptService_InsertSmartArt_OutputMessage** response message.

3.1.4.14.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.14.1.1 IPptEditInternal_InsertSmartArt_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/InsertSmartArt
```

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

3.1.4.14.1.2 IPptEditInternal_InsertSmartArt_OutputMessage

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

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

3.1.4.14.2 Elements

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

3.1.4.14.2.1 InsertSmartArt

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

```
<xs:element name="InsertSmartArt">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="layoutUri" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation into which the SmartArt diagram is inserted, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide into which the SmartArt diagram is inserted.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the placeholder shape into which the SmartArt diagram is inserted.

layoutUri: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the layout with which the SmartArt diagram is inserted.

category: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the category to which the layout belongs.

3.1.4.14.2.2 InsertSmartArtResponse

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

```
<xs:element name="InsertSmartArtResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="InsertSmartArtResult" nillable="true" type="q1:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

InsertSmartArtResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.14.3 Complex Types

None.

3.1.4.14.4 Simple Types

None.

3.1.4.14.5 Attributes

None.

3.1.4.14.6 Groups

None.

3.1.4.14.7 Attribute Groups

None.

3.1.4.15 MoveSlide

The **MoveSlide** operation is used to move a presentation slide in a presentation.

```
<wsdl:operation name="MoveSlide">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/MoveSlide"
    message="tns:IPptEditInternal_MoveSlide_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/MoveSlideResponse"
    message="tns:IPptEditInternal_MoveSlide_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_MoveSlide_InputMessage** request message, and the protocol server responds with an **IPptService_MoveSlide_OutputMessage** response message.

3.1.4.15.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.15.1.1 IPptEditInternal_MoveSlide_InputMessage

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

The SOAP action value is:

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

3.1.4.15.1.2 IPptEditInternal_MoveSlide_OutputMessage

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

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

3.1.4.15.2 Elements

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

3.1.4.15.2.1 MoveSlide

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

```
<xs:element name="MoveSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="srcId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the slide that is moved, as specified in section [3.1.4.11](#).

srcId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide that is moved.

dstId: A value of zero or an **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide that precedes the new location of the moved slide. A value of zero indicates that no presentation slide precedes the new location. In this case, the new location of the moved presentation slide is the first slide of the presentation.

3.1.4.15.2.2 MoveSlideResponse

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

```
<xs:element name="MoveSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q8="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="MoveSlideResult" nillable="true" type="q8:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

MoveSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.15.3 Complex Types

None.

3.1.4.15.4 Simple Types

None.

3.1.4.15.5 Attributes

None.

3.1.4.15.6 Groups

None.

3.1.4.15.7 Attribute Groups

None.

3.1.4.16 Noop

Reserved. MUST not be called.

```
<wsdl:operation name="Noop">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEditTest/Noop" message="tns:IPptEditInternal_Noop_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEditTest/NoopResponse"
    message="tns:IPptEditInternal_Noop_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

Reserved. MUST not be called.

3.1.4.16.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.16.1.1 IPptEditInternal_Noop_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEditTest/Noop
```

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

3.1.4.16.1.2 IPptEditInternal_Noop_OutputMessage

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

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

3.1.4.16.2 Elements

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

3.1.4.16.2.1 Noop

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

```
<xs:element name="Noop">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="fHitBackend" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: Reserved. MUST be ignored.

fHitBackend: Reserved. MUST be ignored.

3.1.4.16.2.2 NoopResponse

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

```
<xs:element name="NoopResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
```

3.1.4.16.3 Complex Types

None.

3.1.4.16.4 Simple Types

None.

3.1.4.16.5 Attributes

None.

3.1.4.16.6 Groups

None.

3.1.4.16.7 Attribute Groups

None.

3.1.4.17 Print

A **Print** operation SHOULD [<16>](#) be used to print a presentation.

```
<wsdl:operation name="Print">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/Print" message="tns:IPptEditInternal_Print_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/PrintResponse"
    message="tns:IPptEditInternal_Print_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_Print_InputMessage** request message, and the protocol server responds with an **IPptService_Print_OutputMessage** response message.

3.1.4.17.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.17.1.1 IPptEditInternal_Print_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/Print
```

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

3.1.4.17.1.2 IPptEditInternal_Print_OutputMessage

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

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

3.1.4.17.2 Elements

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

3.1.4.17.2.1 Print

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

```
<xs:element name="Print">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([XMLSCHEMA2](#) section 3.2.1) that specifies the identifier of the presentation to print, as specified in section [3.1.4.11](#).

3.1.4.17.2.2 PrintResponse

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

```

<xs:element name="PrintResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q21="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="PrintResult" nillable="true" type="q21:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

PrintResult: A **PptViewingService.PrintResult** (section [2.2.4.23](#)) that specifies the result of the operation.

3.1.4.17.3 Complex Types

None.

3.1.4.17.4 Simple Types

None.

3.1.4.17.5 Attributes

None.

3.1.4.17.6 Groups

None.

3.1.4.17.7 Attribute Groups

None.

3.1.4.18 Redo

The **Redo** operation is used to redo the last command in a presentation that is undone.

```

<wsdl:operation name="Redo">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/Redo" message="tns:IPptEditInternal_Redo_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>

```



```

    <wsdl:output
      wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
      ServerInternalService/IPptEdit/RedoResponse"
      message="tns:IPptEditInternal_Redo_OutputMessage"
      xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:operation>

```

The protocol client sends an **IPptService_Redo_InputMessage** request message, and the protocol server responds with an **IPptService_Redo_OutputMessage** response message.

3.1.4.18.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.18.1.1 IPptEditInternal_Redo_InputMessage

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

The SOAP action value is:

```

http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/Redo

```

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

3.1.4.18.1.2 IPptEditInternal_Redo_OutputMessage

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

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

3.1.4.18.2 Elements

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

3.1.4.18.2.1 Redo

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

```

<xs:element name="Redo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

3.1.4.18.2.2 RedoResponse

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

```

<xs:element name="RedoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q17="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="RedoResult" nillable="true" type="q17:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

RedoResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.18.3 Complex Types

None.

3.1.4.18.4 Simple Types

None.

3.1.4.18.5 Attributes

None.

3.1.4.18.6 Groups

None.

3.1.4.18.7 Attribute Groups

None.

3.1.4.19 ReplaceNotes

The **ReplaceNotes** operation is used to replace presentation notes in a presentation.

```

<wsdl:operation name="ReplaceNotes">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceNotes"
message="tns:IPptEditInternal_ReplaceNotes_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceNotesResponse"
message="tns:IPptEditInternal_ReplaceNotes_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ReplaceNotes_InputMessage** request message, and the protocol server responds with an **IPptService_ReplaceNotes_OutputMessage** response message.

3.1.4.19.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.19.1.1 IPptEditInternal_ReplaceNotes_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ReplaceNotes
```

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

3.1.4.19.1.2 IPptEditInternal_ReplaceNotes_OutputMessage

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

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

3.1.4.19.2 Elements

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

3.1.4.19.2.1 ReplaceNotes

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

```
<xs:element name="ReplaceNotes">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="newNotes" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the text that is replaced, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide containing the text that is replaced.

newNotes: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the replacement notes.

3.1.4.19.2.2 ReplaceNotesResponse

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

```
<xs:element name="ReplaceNotesResponse">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element xmlns:q15="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReplaceNotesResult" nillable="true" type="q15:ServiceResult"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

```

ReplaceNotesResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.19.3 Complex Types

None.

3.1.4.19.4 Simple Types

None.

3.1.4.19.5 Attributes

None.

3.1.4.19.6 Groups

None.

3.1.4.19.7 Attribute Groups

None.

3.1.4.20 ReplaceText

The **ReplaceText** operation is used to replace text in a shape in a presentation.

```

<wsdl:operation name="ReplaceText">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceText"
message="tns:IPptEditInternal_ReplaceText_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceTextResponse"
message="tns:IPptEditInternal_ReplaceText_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ReplaceText_InputMessage** request message, and the protocol server responds with an **IPptService_ReplaceText_OutputMessage** response message.

3.1.4.20.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.20.1.1 IPptEditInternal_ReplaceText_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ReplaceText
```

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

3.1.4.20.1.2 IPptEditInternal_ReplaceText_OutputMessage

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

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

3.1.4.20.2 Elements

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

3.1.4.20.2.1 ReplaceText

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

```
<xs:element name="ReplaceText">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="newText" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the text that is replaced, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide containing the text that is replaced.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the shape containing the text that is replaced.

newText: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the replacement text.

3.1.4.20.2.2 ReplaceTextResponse

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

```
<xs:element name="ReplaceTextResponse">
  <xs:complexType>
    <xs:sequence>
```

```

        <xs:element xmlns:q10="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReplaceTextResult" nillable="true" type="q10:ServiceResult"/>
    </xs:sequence>
</xs:complexType>
</xs:element>

```

ReplaceTextResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.20.3 Complex Types

None.

3.1.4.20.4 Simple Types

None.

3.1.4.20.5 Attributes

None.

3.1.4.20.6 Groups

None.

3.1.4.20.7 Attribute Groups

None.

3.1.4.21 ResetSmartArt

The **ResetSmartArt** operation is used to remove all customizations on a SmartArt diagram in a presentation.

```

<wsdl:operation name="ResetSmartArt">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ResetSmartArt"
message="tns:IPptEditInternal_ResetSmartArt_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ResetSmartArtResponse"
message="tns:IPptEditInternal_ResetSmartArt_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>

```

The protocol client sends an **IPptService_ResetSmartArt_InputMessage** request message, and the protocol server responds with an **IPptService_ResetSmartArt_OutputMessage** response message.

3.1.4.21.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.21.1.1 IPptEditInternal_ResetSmartArt_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ResetSmartArt
```

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

3.1.4.21.1.2 IPptEditInternal_ResetSmartArt_OutputMessage

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

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

3.1.4.21.2 Elements

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

3.1.4.21.2.1 ResetSmartArt

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

```
<xs:element name="ResetSmartArt">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the SmartArt diagram that is reset, as specified in section [3.1.4.11](#).

slideId: A **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide containing the SmartArt diagram that is reset.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the SmartArt diagram that is reset.

3.1.4.21.2.2 ResetSmartArtResponse

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

```
<xs:element name="ResetSmartArtResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q2="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ResetSmartArtResult" nillable="true" type="q2:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
```

</xs:element>

ResetSmartArtResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.21.3 Complex Types

None.

3.1.4.21.4 Simple Types

None.

3.1.4.21.5 Attributes

None.

3.1.4.21.6 Groups

None.

3.1.4.21.7 Attribute Groups

None.

3.1.4.22 ReverseSmartArt

The **ReverseSmartArt** operation is used to reverse the direction of a SmartArt diagram in a presentation.

```
<wsdl:operation name="ReverseSmartArt">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ReverseSmartArt"
    message="tns:IPptEditInternal_ReverseSmartArt_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ReverseSmartArtResponse"
    message="tns:IPptEditInternal_ReverseSmartArt_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_ReverseSmartArt_InputMessage** request message, and the protocol server responds with an **IPptService_ReverseSmartArt_OutputMessage** response message.

3.1.4.22.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.22.1.1 IPptEditInternal_ReverseSmartArt_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ReverseSmartArt
```

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

3.1.4.22.1.2 IPptEditInternal_ReverseSmartArt_OutputMessage

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

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

3.1.4.22.2 Elements

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

3.1.4.22.2.1 ReverseSmartArt

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

```
<xs:element name="ReverseSmartArt">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the SmartArt diagram that is reversed, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the presentation slide containing the SmartArt diagram that is reversed.

shapeId: An **ST_DrawingElementId** ([\[ISO/IEC-29500-1\]](#) section 20.1.10.21) that specifies the identifier of the SmartArt diagram that is reversed.

3.1.4.22.2.2 ReverseSmartArtResponse

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

```
<xs:element name="ReverseSmartArtResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q3="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReverseSmartArtResult" nillable="true" type="q3:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ReverseSmartArtResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.22.3 Complex Types

None.

3.1.4.22.4 Simple Types

None.

3.1.4.22.5 Attributes

None.

3.1.4.22.6 Groups

None.

3.1.4.22.7 Attribute Groups

None.

3.1.4.23 SaveAndClose

The **SaveAndClose** operation is used to save the contents of a presentation and unlock it.

```
<wsdl:operation name="SaveAndClose">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/SaveAndClose"
    message="tns:IPptEditInternal_SaveAndClose_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/SaveAndCloseResponse"
    message="tns:IPptEditInternal_SaveAndClose_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_SaveAndClose_InputMessage** request message, and the protocol server responds with an **IPptService_SaveAndClose_OutputMessage** response message.

3.1.4.23.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.23.1.1 IPptEditInternal_SaveAndClose_InputMessage

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

The SOAP action value is:

<http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/SaveAndClose>

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

3.1.4.23.1.2 IPptEditInternal_SaveAndClose_OutputMessage

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

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

3.1.4.23.2 Elements

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

3.1.4.23.2.1 SaveAndClose

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

```
<xs:element name="SaveAndClose">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="fReleaseLock" type="xs:boolean"/>
      <xs:element minOccurs="0" name="fWaitForSave" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

fReleaseLock: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the presentation will become unlocked.

fWaitForSave: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies whether the protocol server makes a blocking call until the save is finished.

3.1.4.23.2.2 SaveAndCloseResponse

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

```
<xs:element name="SaveAndCloseResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q19="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="SaveAndCloseResult" nillable="true" type="q19:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

SaveAndCloseResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.23.3 Complex Types

None.

3.1.4.23.4 Simple Types

None.

3.1.4.23.5 Attributes

None.

3.1.4.23.6 Groups

None.

3.1.4.23.7 Attribute Groups

None.

3.1.4.24 ShowHideSlide

The **ShowHideSlide** operation is used to show or hide a presentation slide.

```
<wsdl:operation name="ShowHideSlide">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/ShowHideSlide"
    message="tns:IPptEditInternal_ShowHideSlide_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    <wsdl:output
      wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
      ServerInternalService/IPptEdit/ShowHideSlideResponse"
      message="tns:IPptEditInternal_ShowHideSlide_OutputMessage"
      xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
    </wsdl:output>
  </wsdl:operation>
```

The protocol client sends an **IPptService_ShowHideSlide_InputMessage** request message, and the protocol server responds with an **IPptService_ShowHideSlide_OutputMessage** response message.

3.1.4.24.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.24.1.1 IPptEditInternal_ShowHideSlide_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/ShowHideSlide
```

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

3.1.4.24.1.2 IPptEditInternal_ShowHideSlide_OutputMessage

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

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

3.1.4.24.2 Elements

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

3.1.4.24.2.1 ShowHideSlide

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

```
<xs:element name="ShowHideSlide">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="fHidden" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation containing the target presentation slide, as specified in section [3.1.4.11](#).

slideId: An **ST_SlideId** ([\[ISO/IEC-29500-1\]](#) section 19.7.13) that specifies the identifier of the target presentation.

fHidden: An **xs:boolean** ([\[XMLSCHEMA2\]](#) section 3.2.2) that specifies if the operation is to show or hide the presentation slide. If the value is true, the protocol server **MUST** make the presentation slide hidden. Otherwise, the protocol server **MUST** make the presentation slide shown. **MUST** be present.

3.1.4.24.2.2 ShowHideSlideResponse

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

```
<xs:element name="ShowHideSlideResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q14="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ShowHideSlideResult" nillable="true" type="q14:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

ShowHideSlideResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element **MUST** be an [EditCommandResponse](#).

3.1.4.24.3 Complex Types

None.

3.1.4.24.4 Simple Types

None.

3.1.4.24.5 Attributes

None.

3.1.4.24.6 Groups

None.

3.1.4.24.7 Attribute Groups

None.

3.1.4.25 Undo

The **Undo** operation is used to undo the last command in a presentation that is not undone.

```
<wsdl:operation name="Undo">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/Undo" message="tns:IPptEditInternal_Undo_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/UndoResponse"
    message="tns:IPptEditInternal_Undo_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

The protocol client sends an **IPptService_Undo_InputMessage** request message, and the protocol server responds with an **IPptService_Undo_OutputMessage** response message.

3.1.4.25.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.25.1.1 IPptEditInternal_Undo_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEdit/Undo
```

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

3.1.4.25.1.2 IPptEditInternal_Undo_OutputMessage

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

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

3.1.4.25.2 Elements

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

3.1.4.25.2.1 Undo

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

```
<xs:element name="Undo">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: An **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the presentation, as specified in section [3.1.4.11](#).

3.1.4.25.2.2 UndoResponse

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

```
<xs:element name="UndoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q16="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="UndoResult" nillable="true" type="q16:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

UndoResult: A [ServiceResult](#) that specifies the result of the operation. The **Result** child element MUST be an [EditCommandResponse](#).

3.1.4.25.3 Complex Types

None.

3.1.4.25.4 Simple Types

None.

3.1.4.25.5 Attributes

None.

3.1.4.25.6 Groups

None.

3.1.4.25.7 Attribute Groups

None.

3.1.4.26 UpdateCanary

Reserved. MUST be ignored.

```
<wsdl:operation name="UpdateCanary">
  <wsdl:input
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/UpdateCanary"
    message="tns:IPptEditInternal_UpdateCanary_InputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
    wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
    ServerInternalService/IPptEdit/UpdateCanaryResponse"
    message="tns:IPptEditInternal_UpdateCanary_OutputMessage"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

3.1.4.26.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.26.1.1 IPptEditInternal_UpdateCanary_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerIntern
alService/IPptEdit/UpdateCanary
```

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

3.1.4.26.1.2 IPptEditInternal_UpdateCanary_OutputMessage

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

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

3.1.4.26.2 Elements

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

3.1.4.26.2.1 UpdateCanary

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

```
<xs:element name="UpdateCanary">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```


presentationId: Reserved. MUST be ignored.

3.1.4.26.2.2 UpdateCanaryResponse

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

```
<xs:element name="UpdateCanaryResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q20="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="UpdateCanaryResult" nillable="true" type="q20:ServiceResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

UpdateCanaryResult: Reserved. MUST be ignored.

3.1.4.26.3 Complex Types

None.

3.1.4.26.4 Simple Types

None.

3.1.4.26.5 Attributes

None.

3.1.4.26.6 Groups

None.

3.1.4.26.7 Attribute Groups

None.

3.1.4.27 WaitCrash

Reserved. MUST not be called.

```
<wsdl:operation name="WaitCrash">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/WaitCrash"
message="tns:IPptEditInternal_WaitCrash_InputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/WaitCrashResponse"
message="tns:IPptEditInternal_WaitCrash_OutputMessage"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"/>
</wsdl:operation>
```

3.1.4.27.1 Messages

The following WSDL message definitions are specific to this operation.

3.1.4.27.1.1 IPptEditInternal_WaitCrash_InputMessage

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

The SOAP action value is:

```
http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServerInternalService/IPptEditTest/WaitCrash
```

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

3.1.4.27.1.2 IPptEditInternal_WaitCrash_OutputMessage

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

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

3.1.4.27.2 Elements

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

3.1.4.27.2.1 WaitCrash

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

```
<xs:element name="WaitCrash">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"/>
      <xs:element minOccurs="0" name="ms" type="xs:unsignedInt"/>
      <xs:element minOccurs="0" name="fCrashWorker" type="xs:boolean"/>
      <xs:element minOccurs="0" name="faultTag" nillable="true" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

presentationId: Reserved. MUST be ignored.

ms: Reserved. MUST be ignored.

fCrashWorker: Reserved. MUST be ignored.

faultTag: Reserved. MUST be ignored.

3.1.4.27.2.2 WaitCrashResponse

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

```
<xs:element name="WaitCrashResponse">
  <xs:complexType>
```

```
<xs:sequence>
  <xs:element xmlns:q25="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="WaitCrashResult" nillable="true" type="q25:ServiceResult"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

WaitCrashResult: Reserved. MUST be ignored.

3.1.4.27.3 Complex Types

None.

3.1.4.27.4 Simple Types

None.

3.1.4.27.5 Attributes

None.

3.1.4.27.6 Groups

None.

3.1.4.27.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

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

4.1 Sample Protocol Interaction

In this example, the protocol client sends a request to the protocol server for the identifier of a presentation. The following **IPptService_GetPresentationId_InputMessage** message is sent to the protocol server:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetPresentationId
      xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
      InternalService/">
      <rawDocId>http://protocolserver/Shared
      Documents/MSPWEDPSEExample.pptx</rawDocId></GetPresentationId>
    </s:Body>
  </s:Envelope>
```

The protocol server responds with the appropriate identifier for the presentation. The following **IPptService_GetPresentationId_OutputMessage** message is sent to the protocol client:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body><GetPresentationIdResponse
    xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
    InternalService/">

    <GetPresentationIdResult>d=F39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a9a007e0723241f
    m568ced57b5cc4aeda74fff96066a8267m</GetPresentationIdResult></GetPresentationIdResponse>
  </s:Body>
</s:Envelope>
```

The response sent to the protocol client also contains the **canary** which is sent back as a **cookie** with the following name: "X-Key". This canary is sent with any subsequent operations to the protocol server that can modify the presentation.

Next, the protocol client sends a request to the protocol server for information about the presentation. The following **IPptService_GetEditPresInfo_InputMessage** message is sent to the protocol server:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body><GetEditPresInfo
    xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
    InternalService/">

    <presentationId>d=F39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a9a007e0723241fm568ced57
    b5cc4aeda74fff96066a8267m</presentationId>
    <firstSlideIndex>0</firstSlideIndex>
    <numSlides>10</numSlides>
    <includedEditSlideIndex>1</includedEditSlideIndex>
  </GetEditPresInfo>
</s:Body>
</s:Envelope>
```

The protocol server responds with the appropriate information about the presentation. The following **IPptService_GetEditPresInfo_OutputMessage** message is sent to the protocol client:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetEditPresInfoResponse
      xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
      InternalService/">
      <GetEditPresInfoResult xmlns:a="http://schemas.datacontract.org/2004/07/p"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Error i:nil="true"/>
        <a:Result i:type="a:EditPresentationInfo">
          <a:EditSlideInfos>
            <a:EditSlideInfoFe>
              <a:Hidden>false</a:Hidden>
              <a:Id>256</a:Id>
              <a:Thumb>
                <a:Alt i:nil="true"/>
                <a:Clip>
                  <a:Bottom>96</a:Bottom>
                  <a:Left>0</a:Left>
                  <a:Right>128</a:Right>
                  <a:Top>0</a:Top>
                </a:Clip>
                <a:Html i:nil="true"/>
                <a:Id>1</a:Id>

                <a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
                9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=14tvmylk%2Ed4e%2Epng</a:ImgId>
                <a:Position>
                  <a:X>0</a:X>
                  <a:Y>0</a:Y>
                </a:Position>
                <a:Z>0</a:Z>
              </a:Thumb>
              <a:Title>Fancy Title</a:Title>
            </a:EditSlideInfoFe>
          </a:EditSlideInfos>
          <a:FirstSlideIndex>0</a:FirstSlideIndex>
          <a:HasComments>false</a:HasComments>
          <a:MainMasters>
            <a:MainMasterFe>
              <a:ContentMasters>
                <a:ContentMasterFe>
                  <a:Id>2147483649</a:Id>
                  <a:Name>Title Slide</a:Name>
                  <a:SlideLayoutType>title</a:SlideLayoutType>
                </a:ContentMasterFe>
                <a:ContentMasterFe>
                  <a:Id>2147483650</a:Id>
                  <a:Name>Title and Content</a:Name>
                  <a:SlideLayoutType>obj</a:SlideLayoutType>
                </a:ContentMasterFe>
                <a:ContentMasterFe>
                  <a:Id>2147483651</a:Id>
                  <a:Name>Section Header</a:Name>
                  <a:SlideLayoutType>secHead</a:SlideLayoutType>
                </a:ContentMasterFe>
                <a:ContentMasterFe>

```

```

        <a:Id>2147483652</a:Id>
        <a:Name>Two Content</a:Name>
        <a:SlideLayoutType>twoObj</a:SlideLayoutType>
    </a:ContentMasterFe>
    <a:ContentMasterFe>
        <a:Id>2147483653</a:Id>
        <a:Name>Comparison</a:Name>
        <a:SlideLayoutType>twoTxTwoObj</a:SlideLayoutType>
    </a:ContentMasterFe>
    <a:ContentMasterFe>
        <a:Id>2147483654</a:Id>
        <a:Name>Title Only</a:Name>
        <a:SlideLayoutType>titleOnly</a:SlideLayoutType>
    </a:ContentMasterFe>
    <a:ContentMasterFe>
        <a:Id>2147483655</a:Id>
        <a:Name>Blank</a:Name>
        <a:SlideLayoutType>blank</a:SlideLayoutType>
    </a:ContentMasterFe>
    <a:ContentMasterFe>
        <a:Id>2147483656</a:Id>
        <a:Name>Content with Caption</a:Name>
        <a:SlideLayoutType>objTx</a:SlideLayoutType>
    </a:ContentMasterFe>
    <a:ContentMasterFe>
        <a:Id>2147483657</a:Id>
        <a:Name>Picture with Caption</a:Name>
        <a:SlideLayoutType>picTx</a:SlideLayoutType>
    </a:ContentMasterFe>
</a:ContentMasters>
<a:Id>2147483648</a:Id>
<a:InContent>true</a:InContent>
<a:Name>Office Theme</a:Name>
</a:MainMasterFe>
</a:MainMasters>
<a:Redo>false</a:Redo>
<a:Rtl>false</a:Rtl>
<a:Size>
    <a:X>620</a:X>
    <a:Y>465</a:Y>
</a:Size>
<a:Slide i:nil="true"/>
<a:SlideOrder>
    <a:SlideIds
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <b:unsignedInt>256</b:unsignedInt>
    </a:SlideIds>
</a:SlideOrder>
<a:Undo>false</a:Undo>
</a:Result>
</GetEditPresInfoResult></GetEditPresInfoResponse>
</s:Body>
</s:Envelope>

```

Next, the protocol client sends a request to the protocol server for information about a presentation slide. The identifier of the first slide is 256 that is specified by the first element of the **SlideOrder** element of the [EditPresentationInfo](#). The following **IPptService_GetEditSlide_InputMessage** message is sent to the protocol server:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body><GetEditSlide
xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
InternalService/"><presentationId>d=F39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a9a007
e0723241fm568ced57b5cc4aeda74fff96066a8267m</presentationId><slideId>256</slideId></GetEditSl
ide></s:Body></s:Envelope>
```

The protocol server responds with the appropriate information about the presentation slide. The following **IPptService_GetEditSlide_OutputMessage** message is sent to the protocol client:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetEditSlideResponse
xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
InternalService/">
      <GetEditSlideResult xmlns:a="http://schemas.datacontract.org/2004/07/p"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Error i:nil="true"/>
        <a:Result i:type="a:EditSlide">
          <a:ContentMasterId>2147483649</a:ContentMasterId>
          <a:EditShapes>
            <a:EditShape>
              <a:Alt i:nil="true"/>
              <a:EditBounds>
                <a:Bottom>140</a:Bottom>
                <a:Left>46</a:Left>
                <a:Right>573</a:Right>
                <a:Top>41</a:Top>
              </a:EditBounds>
              <a:EditHtml>&lt;div class=defaultPHStyles style='font-size:32.0px;font-
family:Calibri;color:#7293C9;text-align:center'&gt; &lt;/div&gt;&lt;p class=nonListPara
style='direction:ltr' lang=en-US style='language:en-US' align=center&gt;&lt;font size=6
face=Calibri color="#7293C9"&gt;&lt;span style='font-variant:normal;text-
transform:uppercase'&gt;&lt;b&gt;Fancy
Title&lt;/b&gt;&lt;/span&gt;&lt;/font&gt;&lt;/p&gt;&lt;/a:EditHtml>
              <a:FontSizeMapping>
                <a:Size1>12</a:Size1>
                <a:Size2>15</a:Size2>
                <a:Size3>19</a:Size3>
                <a:Size4>21</a:Size4>
                <a:Size5>28</a:Size5>
                <a:Size6>44</a:Size6>
                <a:Size7>56</a:Size7>
              </a:FontSizeMapping>
              <a:IsPicture>>false</a:IsPicture>
              <a:IsPlaceholder>>true</a:IsPlaceholder>
              <a:IsPrompt>>false</a:IsPrompt>
              <a:IsSmartArt>>false</a:IsSmartArt>
              <a:PlaceholderType>2</a:PlaceholderType>
              <a:ShapeId>2</a:ShapeId>
              <a:ShapeViewElemId>2</a:ShapeViewElemId>
              <a:SmartArtInfo i:nil="true"/>
              <a:TextViewElemId>3</a:TextViewElemId>
            </a:EditShape>
            <a:EditShape>
              <a:Alt i:nil="true"/>
              <a:EditBounds>
                <a:Bottom>340</a:Bottom>
                <a:Left>25</a:Left>
                <a:Right>459</a:Right>
```

```

    <a:Top>222</a:Top>
  </a:EditBounds>
  <a:EditHtml>&lt;div class=defaultPHStyles style='font-size:24.0px;font-
family:Calibri;color:#1F497D;text-align:center'&gt; &lt;/div&gt;&lt;p class=nonListPara
style='direction:ltr' lang=en-US style='language:en-US' align=center&gt;&lt;font size=5
face=Calibri color="#1F497D"&gt;Boring subtitle&lt;/font&gt;&lt;/p&gt;</a:EditHtml>
  <a:FontSizeMapping>
    <a:Size1>12</a:Size1>
    <a:Size2>20</a:Size2>
    <a:Size3>24</a:Size3>
    <a:Size4>28</a:Size4>
    <a:Size5>32</a:Size5>
    <a:Size6>37</a:Size6>
    <a:Size7>56</a:Size7>
  </a:FontSizeMapping>
  <a:IsPicture>>false</a:IsPicture>
  <a:IsPlaceholder>>true</a:IsPlaceholder>
  <a:IsPrompt>>false</a:IsPrompt>
  <a:IsSmartArt>>false</a:IsSmartArt>
  <a:PlaceholderType>3</a:PlaceholderType>
  <a:ShapeId>3</a:ShapeId>
  <a:ShapeViewElemId>0</a:ShapeViewElemId>
  <a:SmartArtInfo i:nil="true"/>
  <a:TextViewElemId>4</a:TextViewElemId>
</a:EditShape>
</a:EditShapes>
<a:Hidden>>false</a:Hidden>
<a:IsTitleContentMaster>>true</a:IsTitleContentMaster>
<a:MainMasterId>2147483648</a:MainMasterId>
<a:Notes>
  <a:Html>
    <a:Text>&lt;p class=nonListPara style='direction:ltr' lang=en-US
style='language:en-US' align=left&gt;&lt;font face=Calibri&gt;&lt;img width=1 height=10
style='visibility:hidden'&gt;&lt;/font&gt;&lt;/p&gt;</a:Text>
  </a:Html>
  <a:IsPrompt>>true</a:IsPrompt>
</a:Notes>
<a:SlideId>256</a:SlideId>
<a:Theme>
  <a:Accent1>
    <a:b>189</a:b>
    <a:g>129</a:g>
    <a:r>79</a:r>
  </a:Accent1>
  <a:Accent2>
    <a:b>77</a:b>
    <a:g>80</a:g>
    <a:r>192</a:r>
  </a:Accent2>
  <a:Accent3>
    <a:b>89</a:b>
    <a:g>187</a:g>
    <a:r>155</a:r>
  </a:Accent3>
  <a:Accent4>
    <a:b>162</a:b>
    <a:g>100</a:g>
    <a:r>128</a:r>
  </a:Accent4>

```



```

<a:Accent5>
  <a:b>198</a:b>
  <a:g>172</a:g>
  <a:r>75</a:r>
</a:Accent5>
<a:Accent6>
  <a:b>70</a:b>
  <a:g>150</a:g>
  <a:r>247</a:r>
</a:Accent6>
<a:Dark1>
  <a:b>0</a:b>
  <a:g>0</a:g>
  <a:r>0</a:r>
</a:Dark1>
<a:Dark2>
  <a:b>125</a:b>
  <a:g>73</a:g>
  <a:r>31</a:r>
</a:Dark2>
<a:Light1>
  <a:b>255</a:b>
  <a:g>255</a:g>
  <a:r>255</a:r>
</a:Light1>
<a:Light2>
  <a:b>225</a:b>
  <a:g>236</a:g>
  <a:r>238</a:r>
</a:Light2>
</a:Theme>
<a:ThumbnailViewElement>
  <a:Alt i:nil="true"/>
  <a:Clip>
    <a:Bottom>96</a:Bottom>
    <a:Left>0</a:Left>
    <a:Right>128</a:Right>
    <a:Top>0</a:Top>
  </a:Clip>
  <a:Html i:nil="true"/>
  <a:Id>1</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&Rid=14tvmlylk%2Ed4e%2Epng</a:ImgId>
  <a:Position>
    <a:X>0</a:X>
    <a:Y>0</a:Y>
  </a:Position>
  <a:Z>0</a:Z>
</a:ThumbnailViewElement>
<a:ViewElems>
  <a:ViewElement>
    <a:Alt/>
    <a:Clip>
      <a:Bottom>205</a:Bottom>
      <a:Left>0</a:Left>
      <a:Right>407</a:Right>
      <a:Top>0</a:Top>
    </a:Clip>

```

```

    <a:Html i:nil="true"/>
    <a:Id>2</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=yvkqezlr%2Esms%2Epng</a:ImgId>
    <a:Position>
        <a:X>149</a:X>
        <a:Y>8</a:Y>
    </a:Position>
    <a:Z>1</a:Z>
</a:ViewElement>
<a:ViewElement>
    <a:Alt i:nil="true"/>
    <a:Clip>
        <a:Bottom>74</a:Bottom>
        <a:Left>407</a:Left>
        <a:Right>547</a:Right>
        <a:Top>0</a:Top>
    </a:Clip>
    <a:Html i:nil="true"/>
    <a:Id>3</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=yvkqezlr%2Esms%2Epng</a:ImgId>
    <a:Position>
        <a:X>245</a:X>
        <a:Y>66</a:Y>
    </a:Position>
    <a:Z>2</a:Z>
</a:ViewElement>
<a:ViewElement>
    <a:Alt i:nil="true"/>
    <a:Clip>
        <a:Bottom>24</a:Bottom>
        <a:Left>547</a:Left>
        <a:Right>708</a:Right>
        <a:Top>0</a:Top>
    </a:Clip>
    <a:Html i:nil="true"/>
    <a:Id>4</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=yvkqezlr%2Esms%2Epng</a:ImgId>
    <a:Position>
        <a:X>163</a:X>
        <a:Y>232</a:Y>
    </a:Position>
    <a:Z>5</a:Z>
</a:ViewElement>
<a:ViewElement>
    <a:Alt i:nil="true"/>
    <a:Clip i:nil="true"/>
    <a:Html>&lt;div
style="position:absolute;top:0px;left:0px;width:620px;height:465px;background-
color:#FFFFFF"/&gt;</a:Html>
    <a:Id>0</a:Id>
    <a:ImgId i:nil="true"/>
    <a:Position>
        <a:X>0</a:X>
        <a:Y>0</a:Y>

```

```

        </a:Position>
        <a:Z>0</a:Z>
    </a:ViewElement>
</a:ViewElements>
</a:Result>
</GetEditSlideResult>
</GetEditSlideResponse>
</s:Body>
</s:Envelope>

```

Next, the protocol client sends a request to the protocol server to modify the text of a shape on the presentation slide that was referenced in the previous step. The identifier of the shape, 2, is specified by the **SlideId** element of the first element in the **EditShapes** element of the [EditSlide](#). The following **IPptService_ReplaceText_InputMessage** message is sent to the protocol server:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ReplaceText
      xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
      InternalService/">
      <presentationId>d=F39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a9a007e0723241fm568ced57
      b5cc4aeda74fff96066a8267m</presentationId>
      <slideId>256</slideId>
      <shapeId>2</shapeId>
      <newText>&lt;p class=nonListPara style='direction:ltr' lang=en-US style='language:en-
      US' align=center&gt;&lt;font size=6 face=Calibri color="#7293C9"&gt;&lt;span style='font-
      variant:normal;text-transform:uppercase'&gt;&lt;b&gt;Fancy Title with
      3d.&lt;/b&gt;&lt;/span&gt;&lt;/font&gt;&lt;/p&gt;</newText>
    </ReplaceText>
  </s:Body>
</s:Envelope>

```

The protocol server responds with the result of the operation. The following **IPptService_ReplaceText_OutputMessage** message is sent to the protocol client:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <ReplaceTextResponse
      xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
      InternalService/">
      <ReplaceTextResult xmlns:a="http://schemas.datacontract.org/2004/07/p"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Error i:nil="true"/>
        <a:Result i:type="a:EditCommandResponse">
          <a:PresChanges>
            <a:AddedSlideInfos/>
            <a:AddedSlides/>
            <a:MainMasterIds i:nil="true"/>
            <a:ModifiedSlides>
              <a>EditSlideChanges>
                <a:DelElems
                  xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                <a:Elems>
                  <a:ViewElement>
                    <a:Alt/>
                    <a:Clip>

```

```

        <a:Bottom>205</a:Bottom>
        <a:Left>0</a:Left>
        <a:Right>407</a:Right>
        <a:Top>0</a:Top>
    </a:Clip>
    <a:Html i:nil="true"/>
    <a:Id>2</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=iv0q3epq%2Edro%2Epng</a:ImgId>
    <a:Position>
        <a:X>149</a:X>
        <a:Y>8</a:Y>
    </a:Position>
    <a:Z>1</a:Z>
</a:ViewElement>
<a:ViewElement>
    <a:Alt i:nil="true"/>
    <a:Clip>
        <a:Bottom>102</a:Bottom>
        <a:Left>407</a:Left>
        <a:Right>659</a:Right>
        <a:Top>0</a:Top>
    </a:Clip>
    <a:Html i:nil="true"/>
    <a:Id>3</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=iv0q3epq%2Edro%2Epng</a:ImgId>
    <a:Position>
        <a:X>199</a:X>
        <a:Y>55</a:Y>
    </a:Position>
    <a:Z>2</a:Z>
</a:ViewElement>
</a:Elms>
<a:Hidden>>false</a:Hidden>
<a:Id>256</a:Id>
<a:ModifiedShapes>
    <a>EditShape>
        <a:Alt i:nil="true"/>
        <a>EditBounds>
            <a:Bottom>140</a:Bottom>
            <a:Left>46</a:Left>
            <a:Right>573</a:Right>
            <a:Top>41</a:Top>
        </a>EditBounds>
        <a>EditHtml>&lt;div class=defaultPHStyles style='font-size:32.0px;font-
family:Calibri;color:#7293C9;text-align:center'&gt; &lt;/div&gt;&lt;p class=nonListPara
style='direction:ltr' lang=en-US style='language:en-US' align=center&gt;&lt;font size=6
face=Calibri color="#7293C9"&gt;&lt;span style='font-variant:normal;text-
transform:uppercase'&gt;&lt;b&gt;Fancy Title with
3d.&lt;/b&gt;&lt;/span&gt;&lt;/font&gt;&lt;/p&gt;</a>EditHtml>
    <a:FontSizeMapping>
        <a:Size1>12</a:Size1>
        <a:Size2>15</a:Size2>
        <a:Size3>19</a:Size3>
        <a:Size4>21</a:Size4>
        <a:Size5>28</a:Size5>
        <a:Size6>44</a:Size6>

```

```

        <a:Size7>56</a:Size7>
      </a:FontSizeMapping>
      <a:IsPicture>>false</a:IsPicture>
      <a:IsPlaceholder>>true</a:IsPlaceholder>
      <a:IsPrompt>>false</a:IsPrompt>
      <a:IsSmartArt>>false</a:IsSmartArt>
      <a:PlaceholderType>2</a:PlaceholderType>
      <a:ShapeId>2</a:ShapeId>
      <a:ShapeViewElemId>2</a:ShapeViewElemId>
      <a:SmartArtInfo i:nil="true"/>
      <a:TextViewElemId>3</a:TextViewElemId>
    </a>EditShape>
  </a:ModifiedShapes>
  <a:Notes i:nil="true"/>
  <a:Refresh>>false</a:Refresh>
  <a:Thumb>
    <a:Alt i:nil="true"/>
    <a:Clip>
      <a:Bottom>96</a:Bottom>
      <a:Left>659</a:Left>
      <a:Right>787</a:Right>
      <a:Top>0</a:Top>
    </a:Clip>
    <a:Html i:nil="true"/>
    <a:Id>5</a:Id>

<a:ImgId>./editImageHandler.ashx?Pid=d%3DF39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a
9a007e0723241fm568ced57b5cc4aeda74fff96066a8267m&amp;Rid=iv0q3epq%2Edro%2Epng</a:ImgId>
    <a:Position>
      <a:X>0</a:X>
      <a:Y>0</a:Y>
    </a:Position>
    <a:Z>0</a:Z>
  </a:Thumb>
  <a:Title>Fancy Title with 3d.</a:Title>
</a>EditSlideChanges>
</a:ModifiedSlides>
<a:Redo>>false</a:Redo>
<a:SlideOrder i:nil="true"/>
<a:Undo>>true</a:Undo>
</a:PresChanges>
</a:Result>
</ReplaceTextResult>
</ReplaceTextResponse>
</s:Body>
</s:Envelope>

```

Finally, the protocol client sends a request to the protocol server to save the changes made to the presentation and end the session. The following **IPptService_SaveAndClose_InputMessage** message is sent to the protocol server:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SaveAndClose
      xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
      InternalService/">

```

```

<presentationId>d=F39bc64b0ef8e43c493bc0191f7fb5371m371f1619a41e40dd8a9a007e0723241fmc8e360c
9a82447c9a2ce13a5987d49cm</presentationId>
  <fReleaseLock>true</fReleaseLock>
  <fWaitForSave>true</fWaitForSave>
</SaveAndClose>
</s:Body>
</s:Envelope>

```

The protocol server responds with the result of the operation. The following **IPptService_SaveAndClose_OutputMessage** message is sent to the protocol client:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SaveAndCloseResponse
xmlns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditServer
InternalService/">
      <SaveAndCloseResult xmlns:a="http://schemas.datacontract.org/2004/07/p"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a>Error i:nil="true"/>
        <a:Result i:type="a:EditCommandResponse">
          <a:PresChanges i:nil="true"/>
        </a:Result>
      </SaveAndCloseResult>
    </SaveAndCloseResponse>
  </s:Body>
</s:Envelope>

```

5 Security

5.1 Security Considerations for Implementers

The **Html** child element of [ViewElement](#), the **Html** child element of [Notes](#), and the **EditHtml** child element of [EditShape](#) contain [\[HTML\]](#) fragments. HTML fragments are susceptible to **cross-site scripting** attacks. Implementers are advised to sanitize these HTML fragments when generating or processing these elements.

Some operations which specify a change to a presentation such as [DeleteSlide](#) are susceptible to **cross-site request forgery** attacks. Implementers are advised to use a canary [<17>](#) to mitigate operations from this type of attack.

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

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:i0="http://tempuri.org/"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:tns="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEditSe
rverInternalService/" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsd1"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:microsoft="http://schemas.microsoft.com/ws/2005/12/wsd1/contract"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:soap="http://schemas.xmlsoap.org/wsd1/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsd1/soap12/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" name="PptService"
targetNamespace="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPoint
EditServerInternalService/" xmlns:wsd1="http://schemas.xmlsoap.org/wsd1/"
  <wsdl:import namespace="http://tempuri.org/" />
  <wsdl:types>
    <xs:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPoint
EditServerInternalService/" xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="ResetSmartArt">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ResetSmartArtResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q1="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ResetSmartArtResult" nillable="true" type="q1:ServiceResult" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ReverseSmartArt">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ReverseSmartArtResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q2="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReverseSmartArtResult" nillable="true" type="q2:ServiceResult" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```



```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetPresentationId">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="rawDocId" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetPresentationIdResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="GetPresentationIdResult" nillable="true"
type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetEditPresInfo">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="firstSlideIndex" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="numSlides" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="includedEditSlideIndex" type="xs:int" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetEditPresInfoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q3="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="GetEditPresInfoResult" nillable="true" type="q3:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetEditSlide">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:int" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetEditSlideResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q4="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="GetEditSlideResult" nillable="true" type="q4:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="MoveSlide">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>

```

```

        <xs:element minOccurs="0" name="srcId" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="MoveSlideResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q5="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="MoveSlideResult" nillable="true" type="q5:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="InsertSlide">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="mainMasterId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="contentMasterId" type="xs:unsignedInt" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="InsertSlideResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q6="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="InsertSlideResult" nillable="true" type="q6:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ReplaceText">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="newText" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ReplaceTextResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q7="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReplaceTextResult" nillable="true" type="q7:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ClearPlaceholder">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        </xs:complexType>
      </xs:element>
      <xs:element name="ClearPlaceholderResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q8="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ClearPlaceholderResult" nillable="true" type="q8:ServiceResult" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeleteSlide">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DeleteSlideResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q9="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="DeleteSlideResult" nillable="true" type="q9:ServiceResult" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DuplicateSlide">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="srcId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="dstId" type="xs:unsignedInt" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="DuplicateSlideResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element xmlns:q10="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="DuplicateSlideResult" nillable="true" type="q10:ServiceResult" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ShowHideSlide">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="fHidden" type="xs:boolean" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ShowHideSlideResponse">
        <xs:complexType>
          <xs:sequence>

```

```

        <xs:element xmlns:q11="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ShowHideSlideResult" nillable="true" type="q11:ServiceResult" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ReplaceNotes">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="newNotes" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ReplaceNotesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q12="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ReplaceNotesResult" nillable="true" type="q12:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Undo">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="UndoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q13="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="UndoResult" nillable="true" type="q13:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Redo">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="RedoResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q14="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="RedoResult" nillable="true" type="q14:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ChangePictureStyle">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="pictureStyleId" type="xs:unsignedInt" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ChangePictureStyleResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q15="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangePictureStyleResult" nillable="true" type="q15:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SaveAndClose">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="fReleaseLock" type="xs:boolean" />
            <xs:element minOccurs="0" name="fWaitForSave" type="xs:boolean" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SaveAndCloseResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q16="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="SaveAndCloseResult" nillable="true" type="q16:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="UpdateCanary">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="UpdateCanaryResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q17="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="UpdateCanaryResult" nillable="true" type="q17:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ChangeSmartArtLayout">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="layoutUri" nillable="true" type="xs:string" />
            <xs:element minOccurs="0" name="category" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>

```

```

</xs:element>
<xs:element name="ChangeSmartArtLayoutResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q18="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtLayoutResult" nillable="true" type="q18:ServiceResult" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ChangeSmartArtColor">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="colorTransUri" nillable="true" type="xs:string"
/>
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ChangeSmartArtColorResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q19="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtColorResult" nillable="true" type="q19:ServiceResult" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ChangeSmartArtStyle">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="styleUri" nillable="true" type="xs:string" />
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ChangeSmartArtStyleResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element xmlns:q20="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="ChangeSmartArtStyleResult" nillable="true" type="q20:ServiceResult" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="InsertSmartArt">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
      <xs:element minOccurs="0" name="slideId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="shapeId" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="layoutUri" nillable="true" type="xs:string" />
      <xs:element minOccurs="0" name="category" nillable="true" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="InsertSmartArtResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q21="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="InsertSmartArtResult" nillable="true" type="q21:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="WaitCrash">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="ms" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="fCrashWorker" type="xs:boolean" />
            <xs:element minOccurs="0" name="faultTag" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="WaitCrashResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element xmlns:q22="http://schemas.datacontract.org/2004/07/p" minOccurs="0"
name="WaitCrashResult" nillable="true" type="q22:ServiceResult" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Noop">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="presentationId" nillable="true" type="xs:string"
/>
            <xs:element minOccurs="0" name="fHitBackend" type="xs:boolean" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="NoopResponse">
    <xs:complexType>
        <xs:sequence />
    </xs:complexType>
</xs:element>
</xs:schema>
<xs:schema xmlns:tns="http://schemas.datacontract.org/2004/07/p"
elementFormDefault="qualified" targetNamespace="http://schemas.datacontract.org/2004/07/p"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="EditCommandResponse">
        <xs:sequence>
            <xs:element minOccurs="0" name="PresChanges" nillable="true"
type="tns:EditPresentationChanges" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="EditCommandResponse" nillable="true" type="tns:EditCommandResponse"
/>
    <xs:complexType name="EditPresentationChanges">
        <xs:sequence>
            <xs:element minOccurs="0" name="AddedSlideInfos" nillable="true"
type="tns:ArrayOfEditSlideInfoFe" />

```

```

        <xs:element minOccurs="0" name="AddedSlides" nillable="true"
type="tns:ArrayOfEditSlide" />
        <xs:element minOccurs="0" name="MainMasterIds" nillable="true"
type="tns:SlideIdList" />
        <xs:element minOccurs="0" name="ModifiedSlides" nillable="true"
type="tns:ArrayOfEditSlideChanges" />
        <xs:element minOccurs="0" name="Redo" type="xs:boolean" />
        <xs:element minOccurs="0" name="SlideOrder" nillable="true" type="tns:SlideIdList"
/>

        <xs:element minOccurs="0" name="Undo" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:element name="EditPresentationChanges" nillable="true"
type="tns:EditPresentationChanges" />
<xs:complexType name="ArrayOfEditSlideInfoFe">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="EditSlideInfoFe"
nillable="true" type="tns:EditSlideInfoFe" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfEditSlideInfoFe" nillable="true"
type="tns:ArrayOfEditSlideInfoFe" />
<xs:complexType name="EditSlideInfoFe">
    <xs:sequence>
        <xs:element minOccurs="0" name="Hidden" type="xs:boolean" />
        <xs:element minOccurs="0" name="Id" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="Thumb" nillable="true" type="tns:ViewElement" />
        <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:element name="EditSlideInfoFe" nillable="true" type="tns:EditSlideInfoFe" />
<xs:complexType name="ViewElement">
    <xs:sequence>
        <xs:element minOccurs="0" name="Alt" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="Clip" nillable="true" type="tns:Rectangle" />
        <xs:element minOccurs="0" name="Html" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="Id" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="ImgId" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="Position" nillable="true" type="tns:Point" />
        <xs:element minOccurs="0" name="Z" type="xs:unsignedInt" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ViewElement" nillable="true" type="tns:ViewElement" />
<xs:complexType name="Rectangle">
    <xs:sequence>
        <xs:element minOccurs="0" name="Bottom" type="xs:int" />
        <xs:element minOccurs="0" name="Left" type="xs:int" />
        <xs:element minOccurs="0" name="Right" type="xs:int" />
        <xs:element minOccurs="0" name="Top" type="xs:int" />
    </xs:sequence>
</xs:complexType>
<xs:element name="Rectangle" nillable="true" type="tns:Rectangle" />
<xs:complexType name="Point">
    <xs:sequence>
        <xs:element minOccurs="0" name="X" type="xs:int" />
        <xs:element minOccurs="0" name="Y" type="xs:int" />
    </xs:sequence>
</xs:complexType>
<xs:element name="Point" nillable="true" type="tns:Point" />

```



```

<xs:complexType name="ArrayOfEditSlide">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="EditSlide" nillable="true"
type="tns:EditSlide" />
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfEditSlide" nillable="true" type="tns:ArrayOfEditSlide" />
<xs:complexType name="EditSlide">
  <xs:sequence>
    <xs:element minOccurs="0" name="ContentMasterId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="EditShapes" nillable="true"
type="tns:ArrayOfEditShape" />
    <xs:element minOccurs="0" name="Hidden" type="xs:boolean" />
    <xs:element minOccurs="0" name="IsTitleContentMaster" type="xs:boolean" />
    <xs:element minOccurs="0" name="MainMasterId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Notes" nillable="true" type="tns:Notes" />
    <xs:element minOccurs="0" name="SlideId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Theme" nillable="true" type="tns:Theme" />
    <xs:element minOccurs="0" name="ThumbnailViewElement" nillable="true"
type="tns:ViewElement" />
    <xs:element minOccurs="0" name="ViewElems" nillable="true"
type="tns:ArrayOfViewElement" />
  </xs:sequence>
</xs:complexType>
<xs:element name="EditSlide" nillable="true" type="tns:EditSlide" />
<xs:complexType name="ArrayOfEditShape">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="EditShape" nillable="true"
type="tns:EditShape" />
  </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfEditShape" nillable="true" type="tns:ArrayOfEditShape" />
<xs:complexType name="EditShape">
  <xs:sequence>
    <xs:element minOccurs="0" name="Alt" nillable="true" type="xs:string" />
    <xs:element minOccurs="0" name="EditBounds" nillable="true" type="tns:Rectangle" />
    <xs:element minOccurs="0" name="EditHtml" nillable="true" type="xs:string" />
    <xs:element minOccurs="0" name="FontSizeMapping" nillable="true"
type="tns:FontSizeMapping" />
    <xs:element minOccurs="0" name="IsPicture" type="xs:boolean" />
    <xs:element minOccurs="0" name="IsPlaceholder" type="xs:boolean" />
    <xs:element minOccurs="0" name="IsPrompt" type="xs:boolean" />
    <xs:element minOccurs="0" name="IsSmartArt" type="xs:boolean" />
    <xs:element minOccurs="0" name="PlaceholderType" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="ShapeId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="ShapeViewElemId" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="SmartArtInfo" nillable="true"
type="tns:SmartArtInfo" />
    <xs:element minOccurs="0" name="TextViewElemId" type="xs:unsignedInt" />
  </xs:sequence>
</xs:complexType>
<xs:element name="EditShape" nillable="true" type="tns:EditShape" />
<xs:complexType name="FontSizeMapping">
  <xs:sequence>
    <xs:element minOccurs="0" name="Size1" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Size2" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Size3" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Size4" type="xs:unsignedInt" />
    <xs:element minOccurs="0" name="Size5" type="xs:unsignedInt" />
  </xs:sequence>
</xs:complexType>

```

```

        <xs:element minOccurs="0" name="Size6" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="Size7" type="xs:unsignedInt" />
    </xs:sequence>
</xs:complexType>
<xs:element name="FontSizeMapping" nillable="true" type="tns:FontSizeMapping" />
<xs:complexType name="SmartArtInfo">
    <xs:sequence>
        <xs:element minOccurs="0" name="fReverseChecked" type="xs:boolean" />
        <xs:element minOccurs="0" name="fReverseEnabled" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:element name="SmartArtInfo" nillable="true" type="tns:SmartArtInfo" />
<xs:complexType name="Notes">
    <xs:sequence>
        <xs:element minOccurs="0" name="Html" nillable="true" type="tns:Html" />
        <xs:element minOccurs="0" name="IsPrompt" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:element name="Notes" nillable="true" type="tns:Notes" />
<xs:complexType name="Html">
    <xs:sequence>
        <xs:element minOccurs="0" name="Text" nillable="true" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:element name="Html" nillable="true" type="tns:Html" />
<xs:complexType name="Theme">
    <xs:sequence>
        <xs:element minOccurs="0" name="Accent1" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Accent2" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Accent3" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Accent4" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Accent5" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Accent6" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Dark1" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Dark2" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Light1" nillable="true" type="tns:ThemeColor" />
        <xs:element minOccurs="0" name="Light2" nillable="true" type="tns:ThemeColor" />
    </xs:sequence>
</xs:complexType>
<xs:element name="Theme" nillable="true" type="tns:Theme" />
<xs:complexType name="ThemeColor">
    <xs:sequence>
        <xs:element minOccurs="0" name="b" type="xs:unsignedByte" />
        <xs:element minOccurs="0" name="g" type="xs:unsignedByte" />
        <xs:element minOccurs="0" name="r" type="xs:unsignedByte" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ThemeColor" nillable="true" type="tns:ThemeColor" />
<xs:complexType name="ArrayOfViewElement">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="ViewElement" nillable="true"
type="tns:ViewElement" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfViewElement" nillable="true" type="tns:ArrayOfViewElement" />
<xs:complexType name="SlideIdList">
    <xs:sequence>
        <xs:element xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="SlideIds" nillable="true" type="q1:ArrayOfunsignedInt" />

```

```

        </xs:sequence>
    </xs:complexType>
    <xs:element name="SlideIdList" nillable="true" type="tns:SlideIdList" />
    <xs:complexType name="ArrayOfEditSlideChanges">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="EditSlideChanges"
nillable="true" type="tns:EditSlideChanges" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ArrayOfEditSlideChanges" nillable="true"
type="tns:ArrayOfEditSlideChanges" />
    <xs:complexType name="EditSlideChanges">
        <xs:sequence>
            <xs:element xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
minOccurs="0" name="DelElems" nillable="true" type="q2:ArrayOfunsignedInt" />
            <xs:element minOccurs="0" name="Elems" nillable="true"
type="tns:ArrayOfViewElement" />
            <xs:element minOccurs="0" name="Hidden" type="xs:boolean" />
            <xs:element minOccurs="0" name="Id" type="xs:unsignedInt" />
            <xs:element minOccurs="0" name="ModifiedShapes" nillable="true"
type="tns:ArrayOfEditShape" />
            <xs:element minOccurs="0" name="Notes" nillable="true" type="tns:Notes" />
            <xs:element minOccurs="0" name="Refresh" type="xs:boolean" />
            <xs:element minOccurs="0" name="Thumb" nillable="true" type="tns:ViewElement" />
            <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="EditSlideChanges" nillable="true" type="tns:EditSlideChanges" />
    <xs:complexType name="ServiceResult">
        <xs:sequence>
            <xs:element minOccurs="0" name="Error" nillable="true" type="tns:ServiceError" />
            <xs:element minOccurs="0" name="Result" nillable="true" type="xs:anyType" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ServiceResult" nillable="true" type="tns:ServiceResult" />
    <xs:complexType name="ServiceError">
        <xs:sequence>
            <xs:element minOccurs="0" name="Message" nillable="true" type="xs:string" />
            <xs:element
xmlns:q3="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Powerpoint.Pipe.Int
erface" minOccurs="0" name="RecommendedActions" type="q3:ClientActions" />
            <xs:element minOccurs="0" name="Title" nillable="true" type="xs:string" />
            <xs:element minOccurs="0" name="Type" type="tns:ServiceErrorType" />
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ServiceError" nillable="true" type="tns:ServiceError" />
    <xs:simpleType name="ServiceErrorType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="UnknownError" />
            <xs:enumeration value="ApplicationError" />
            <xs:enumeration value="Timeout" />
            <xs:enumeration value="ServiceBusy" />
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="ServiceErrorType" nillable="true" type="tns:ServiceErrorType" />
    <xs:complexType name="EditPresentationInfo">
        <xs:sequence>
            <xs:element minOccurs="0" name="EditSlideInfos" nillable="true"
type="tns:ArrayOfEditSlideInfoFe" />
            <xs:element minOccurs="0" name="FirstSlideIndex" type="xs:unsignedInt" />

```

```

        <xs:element minOccurs="0" name="HasComments" type="xs:boolean" />
        <xs:element minOccurs="0" name="MainMasters" nillable="true"
type="tns:ArrayOfMainMasterFe" />
        <xs:element minOccurs="0" name="Redo" type="xs:boolean" />
        <xs:element minOccurs="0" name="Rtl" type="xs:boolean" />
        <xs:element minOccurs="0" name="Size" nillable="true" type="tns:Point" />
        <xs:element minOccurs="0" name="Slide" nillable="true" type="tns:EditSlide" />
        <xs:element minOccurs="0" name="SlideOrder" nillable="true" type="tns:SlideIdList"
/>
        <xs:element minOccurs="0" name="Undo" type="xs:boolean" />
    </xs:sequence>
</xs:complexType>
<xs:element name="EditPresentationInfo" nillable="true" type="tns:EditPresentationInfo"
/>
<xs:complexType name="ArrayOfMainMasterFe">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="MainMasterFe" nillable="true"
type="tns:MainMasterFe" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfMainMasterFe" nillable="true" type="tns:ArrayOfMainMasterFe"
/>
<xs:complexType name="MainMasterFe">
    <xs:sequence>
        <xs:element minOccurs="0" name="ContentMasters" nillable="true"
type="tns:ArrayOfContentMasterFe" />
        <xs:element minOccurs="0" name="Id" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="InContent" type="xs:boolean" />
        <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:element name="MainMasterFe" nillable="true" type="tns:MainMasterFe" />
<xs:complexType name="ArrayOfContentMasterFe">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="ContentMasterFe"
nillable="true" type="tns:ContentMasterFe" />
    </xs:sequence>
</xs:complexType>
<xs:element name="ArrayOfContentMasterFe" nillable="true"
type="tns:ArrayOfContentMasterFe" />
<xs:complexType name="ContentMasterFe">
    <xs:sequence>
        <xs:element minOccurs="0" name="Id" type="xs:unsignedInt" />
        <xs:element minOccurs="0" name="Name" nillable="true" type="xs:string" />
        <xs:element minOccurs="0" name="SlideLayoutType" nillable="true" type="xs:string"
/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ContentMasterFe" nillable="true" type="tns:ContentMasterFe" />
</xs:schema>
<xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="ArrayOfunsignedInt">
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="unsignedInt"
type="xs:unsignedInt" />
        </xs:sequence>
    </xs:complexType>

```

```

        <xs:element name="ArrayOfunsignedInt" nillable="true" type="tns:ArrayOfunsignedInt" />
    </xs:schema>
    <xs:schema
        xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Powerpoint.Pipe.Interface"
        elementFormDefault="qualified"
        targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.Office.Server.Powerpoint.Pipe.Interface"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:simpleType name="ClientActions">
            <xs:list>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="None">
                            <xs:annotation>
                                <xs:appinfo>
                                    <EnumerationValue
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">0</EnumerationValue>
                                </xs:appinfo>
                            </xs:annotation>
                        </xs:enumeration>
                        <xs:enumeration value="Dismiss">
                            <xs:annotation>
                                <xs:appinfo>
                                    <EnumerationValue
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">1</EnumerationValue>
                                </xs:appinfo>
                            </xs:annotation>
                        </xs:enumeration>
                        <xs:enumeration value="Close">
                            <xs:annotation>
                                <xs:appinfo>
                                    <EnumerationValue
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">2</EnumerationValue>
                                </xs:appinfo>
                            </xs:annotation>
                        </xs:enumeration>
                        <xs:enumeration value="OpenInClient">
                            <xs:annotation>
                                <xs:appinfo>
                                    <EnumerationValue
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">4</EnumerationValue>
                                </xs:appinfo>
                            </xs:annotation>
                        </xs:enumeration>
                        <xs:enumeration value="Refresh">
                            <xs:annotation>
                                <xs:appinfo>
                                    <EnumerationValue
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">8</EnumerationValue>
                                </xs:appinfo>
                            </xs:annotation>
                        </xs:enumeration>
                    </xs:restriction>
                </xs:simpleType>
            </xs:list>
        </xs:simpleType>
        <xs:element name="ClientActions" nillable="true" type="tns:ClientActions" />
    </xs:schema>
    <xs:schema xmlns:tns="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="tns:char" />
  <xs:simpleType name="char">
    <xs:restriction base="xs:int" />
  </xs:simpleType>
  <xs:element name="duration" nillable="true" type="tns: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="tns: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>
</wsdl:types>
<wsdl:message name="IPptEditInternal_ResetSmartArt_InputMessage">
  <wsdl:part name="parameters" element="tns:ResetSmartArt" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ResetSmartArt_OutputMessage">
  <wsdl:part name="parameters" element="tns:ResetSmartArtResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ReverseSmartArt_InputMessage">
  <wsdl:part name="parameters" element="tns:ReverseSmartArt" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ReverseSmartArt_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReverseSmartArtResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetPresentationId_InputMessage">
  <wsdl:part name="parameters" element="tns:GetPresentationId" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetPresentationId_OutputMessage">

```

```

    <wsdl:part name="parameters" element="tns:GetPresentationIdResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetEditPresInfo_InputMessage">
    <wsdl:part name="parameters" element="tns:GetEditPresInfo" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetEditPresInfo_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetEditPresInfoResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetEditSlide_InputMessage">
    <wsdl:part name="parameters" element="tns:GetEditSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_GetEditSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns:GetEditSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_MoveSlide_InputMessage">
    <wsdl:part name="parameters" element="tns:MoveSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_MoveSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns:MoveSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_InsertSlide_InputMessage">
    <wsdl:part name="parameters" element="tns:InsertSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_InsertSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns:InsertSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ReplaceText_InputMessage">
    <wsdl:part name="parameters" element="tns:ReplaceText" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ReplaceText_OutputMessage">
    <wsdl:part name="parameters" element="tns:ReplaceTextResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ClearPlaceholder_InputMessage">
    <wsdl:part name="parameters" element="tns:ClearPlaceholder" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ClearPlaceholder_OutputMessage">
    <wsdl:part name="parameters" element="tns:ClearPlaceholderResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_DeleteSlide_InputMessage">
    <wsdl:part name="parameters" element="tns>DeleteSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_DeleteSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns>DeleteSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_DuplicateSlide_InputMessage">
    <wsdl:part name="parameters" element="tns:DuplicateSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_DuplicateSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns:DuplicateSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ShowHideSlide_InputMessage">
    <wsdl:part name="parameters" element="tns:ShowHideSlide" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ShowHideSlide_OutputMessage">
    <wsdl:part name="parameters" element="tns:ShowHideSlideResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ReplaceNotes_InputMessage">
    <wsdl:part name="parameters" element="tns:ReplaceNotes" />
</wsdl:message>

```

```

<wsdl:message name="IPptEditInternal_ReplaceNotes_OutputMessage">
  <wsdl:part name="parameters" element="tns:ReplaceNotesResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Undo_InputMessage">
  <wsdl:part name="parameters" element="tns:Undo" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Undo_OutputMessage">
  <wsdl:part name="parameters" element="tns:UndoResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Redo_InputMessage">
  <wsdl:part name="parameters" element="tns:Redo" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Redo_OutputMessage">
  <wsdl:part name="parameters" element="tns:RedoResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangePictureStyle_InputMessage">
  <wsdl:part name="parameters" element="tns:ChangePictureStyle" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangePictureStyle_OutputMessage">
  <wsdl:part name="parameters" element="tns:ChangePictureStyleResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_SaveAndClose_InputMessage">
  <wsdl:part name="parameters" element="tns:SaveAndClose" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_SaveAndClose_OutputMessage">
  <wsdl:part name="parameters" element="tns:SaveAndCloseResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_UpdateCanary_InputMessage">
  <wsdl:part name="parameters" element="tns:UpdateCanary" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_UpdateCanary_OutputMessage">
  <wsdl:part name="parameters" element="tns:UpdateCanaryResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtLayout_InputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtLayout" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtLayout_OutputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtLayoutResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtColor_InputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtColor" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtColor_OutputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtColorResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtStyle_InputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtStyle" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_ChangeSmartArtStyle_OutputMessage">
  <wsdl:part name="parameters" element="tns:ChangeSmartArtStyleResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_InsertSmartArt_InputMessage">
  <wsdl:part name="parameters" element="tns:InsertSmartArt" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_InsertSmartArt_OutputMessage">
  <wsdl:part name="parameters" element="tns:InsertSmartArtResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_WaitCrash_InputMessage">
  <wsdl:part name="parameters" element="tns:WaitCrash" />

```



```

</wsdl:message>
<wsdl:message name="IPptEditInternal_WaitCrash_OutputMessage">
  <wsdl:part name="parameters" element="tns:WaitCrashResponse" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Noop_InputMessage">
  <wsdl:part name="parameters" element="tns:Noop" />
</wsdl:message>
<wsdl:message name="IPptEditInternal_Noop_OutputMessage">
  <wsdl:part name="parameters" element="tns:NoopResponse" />
</wsdl:message>
<wsdl:portType name="IPptEditInternal">
  <wsdl:operation name="ResetSmartArt">
    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ResetSmartArt"
message="tns:IPptEditInternal_ResetSmartArt_InputMessage" />
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ResetSmartArtResponse"
message="tns:IPptEditInternal_ResetSmartArt_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="ReverseSmartArt">
      <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReverseSmartArt"
message="tns:IPptEditInternal_ReverseSmartArt_InputMessage" />
      <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReverseSmartArtResponse"
message="tns:IPptEditInternal_ReverseSmartArt_OutputMessage" />
      </wsdl:operation>
      <wsdl:operation name="GetPresentationId">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetPresentationId"
message="tns:IPptEditInternal_GetPresentationId_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetPresentationIdResponse"
message="tns:IPptEditInternal_GetPresentationId_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="GetEditPresInfo">
          <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditPresInfo"
message="tns:IPptEditInternal_GetEditPresInfo_InputMessage" />
          <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditPresInfoResponse"
message="tns:IPptEditInternal_GetEditPresInfo_OutputMessage" />
          </wsdl:operation>
          <wsdl:operation name="GetEditSlide">
            <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditSlide"
message="tns:IPptEditInternal_GetEditSlide_InputMessage" />
            <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/GetEditSlideResponse"
message="tns:IPptEditInternal_GetEditSlide_OutputMessage" />
            </wsdl:operation>
            <wsdl:operation name="MoveSlide">

```

```

        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/MoveSlide"
message="tns:IPptEditInternal_MoveSlide_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/MoveSlideResponse"
message="tns:IPptEditInternal_MoveSlide_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="InsertSlide">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/InsertSlide"
message="tns:IPptEditInternal_InsertSlide_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/InsertSlideResponse"
message="tns:IPptEditInternal_InsertSlide_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="ReplaceText">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceText"
message="tns:IPptEditInternal_ReplaceText_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceTextResponse"
message="tns:IPptEditInternal_ReplaceText_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="ClearPlaceholder">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ClearPlaceholder"
message="tns:IPptEditInternal_ClearPlaceholder_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ClearPlaceholderResponse"
message="tns:IPptEditInternal_ClearPlaceholder_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="DeleteSlide">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DeleteSlide"
message="tns:IPptEditInternal_DeleteSlide_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DeleteSlideResponse"
message="tns:IPptEditInternal_DeleteSlide_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="DuplicateSlide">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DuplicateSlide"
message="tns:IPptEditInternal_DuplicateSlide_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/DuplicateSlideResponse"
message="tns:IPptEditInternal_DuplicateSlide_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="ShowHideSlide">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit

```

```

ServerInternalService/IPptEdit/ShowHideSlide"
message="tns:IPptEditInternal_ShowHideSlide_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ShowHideSlideResponse"
message="tns:IPptEditInternal_ShowHideSlide_OutputMessage" />
  </wsdl:operation>
  <wsdl:operation name="ReplaceNotes">
    <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceNotes"
message="tns:IPptEditInternal_ReplaceNotes_InputMessage" />
    <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ReplaceNotesResponse"
message="tns:IPptEditInternal_ReplaceNotes_OutputMessage" />
    </wsdl:operation>
    <wsdl:operation name="Undo">
      <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/Undo" message="tns:IPptEditInternal_Undo_InputMessage" />
      <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/UndoResponse"
message="tns:IPptEditInternal_Undo_OutputMessage" />
      </wsdl:operation>
      <wsdl:operation name="Redo">
        <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/Redo" message="tns:IPptEditInternal_Redo_InputMessage" />
        <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/RedoResponse"
message="tns:IPptEditInternal_Redo_OutputMessage" />
        </wsdl:operation>
        <wsdl:operation name="ChangePictureStyle">
          <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangePictureStyle"
message="tns:IPptEditInternal_ChangePictureStyle_InputMessage" />
          <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangePictureStyleResponse"
message="tns:IPptEditInternal_ChangePictureStyle_OutputMessage" />
          </wsdl:operation>
          <wsdl:operation name="SaveAndClose">
            <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/SaveAndClose"
message="tns:IPptEditInternal_SaveAndClose_InputMessage" />
            <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/SaveAndCloseResponse"
message="tns:IPptEditInternal_SaveAndClose_OutputMessage" />
            </wsdl:operation>
            <wsdl:operation name="UpdateCanary">
              <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/UpdateCanary"
message="tns:IPptEditInternal_UpdateCanary_InputMessage" />
              <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit

```

```

ServerInternalService/IPptEdit/UpdateCanaryResponse"
message="tns:IPptEditInternal_UpdateCanary_OutputMessage" />
</wsdl:operation>
<wsdl:operation name="ChangeSmartArtLayout">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtLayout"
message="tns:IPptEditInternal_ChangeSmartArtLayout_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtLayoutResponse"
message="tns:IPptEditInternal_ChangeSmartArtLayout_OutputMessage" />
  </wsdl:operation>
<wsdl:operation name="ChangeSmartArtColor">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtColor"
message="tns:IPptEditInternal_ChangeSmartArtColor_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtColorResponse"
message="tns:IPptEditInternal_ChangeSmartArtColor_OutputMessage" />
  </wsdl:operation>
<wsdl:operation name="ChangeSmartArtStyle">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtStyle"
message="tns:IPptEditInternal_ChangeSmartArtStyle_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/ChangeSmartArtStyleResponse"
message="tns:IPptEditInternal_ChangeSmartArtStyle_OutputMessage" />
  </wsdl:operation>
<wsdl:operation name="InsertSmartArt">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/InsertSmartArt"
message="tns:IPptEditInternal_InsertSmartArt_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEdit/InsertSmartArtResponse"
message="tns:IPptEditInternal_InsertSmartArt_OutputMessage" />
  </wsdl:operation>
<wsdl:operation name="WaitCrash">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/WaitCrash"
message="tns:IPptEditInternal_WaitCrash_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/WaitCrashResponse"
message="tns:IPptEditInternal_WaitCrash_OutputMessage" />
  </wsdl:operation>
<wsdl:operation name="Noop">
  <wsdl:input
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/Noop" message="tns:IPptEditInternal_Noop_InputMessage" />
  <wsdl:output
wsaw:Action="http://schemas.microsoft.com/office/PowerPoint/Server/WebServices/PowerPointEdit
ServerInternalService/IPptEditTest/NoopResponse"
message="tns:IPptEditInternal_Noop_OutputMessage" />
  </wsdl:operation>
</wsdl:portType>

```

</wsdl:definitions>

7 Appendix B: Product Behavior

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

- Microsoft® PowerPoint® 2010
- Microsoft® PowerPoint® Web App

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

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

<1> [Section 2.1](#): PowerPoint Web App servers support JSON [\[RFC4627\]](#) over HTTP and HTTPS.

<2> [Section 2.2.4](#): Support for this type was added in Office 2010 Service Pack 1.

<3> [Section 2.2.4](#): Support for this type was added in Office 2010 Service Pack 1.

<4> [Section 2.2.4.9](#): Support for this type was added in Office 2010 Service Pack 1.

<5> [Section 2.2.4.12](#): Support for this element was added in Office 2010 Service Pack 1.

<6> [Section 2.2.4.12](#): Support for this element was added in Office 2010 Service Pack 1.

<7> [Section 2.2.4.13](#): Support for this element was added in Office 2010 Service Pack 1.

<8> [Section 2.2.4.23](#): Support for this type was added in Office 2010 Service Pack 1.

<9> [Section 2.2.4.31](#): PowerPoint Web App defines this type as an **xs:string** ([\[XMLSCHEMA2\]](#) section 3.2.1) that specifies the identifier of the image.

To retrieve an image, the protocol client sends a request to the protocol server at the endpoint formed by appending `"/_layouts/editImageHandler.ashx"` to the URL of the Web site, for example `http://www.contoso.com/Repository/_layouts/editImageHandler.ashx`.

The parameters to the endpoint are specified by the following table:

Parameter name	Definition
Rid	MUST be <code>ImgId</code> as specified by ViewElement .
Pid	MUST be a valid identifier for a presentation as specified in GetPresentationId .

<10> [Section 3.1.4](#): The insert picture operation is used to insert a picture into a presentation.

The protocol server endpoint to insert picture content is formed by appending `"/_layouts/pptInsertPicture.ashx"` to the URL of the Web site, for example `http://www.contoso.com/Repository/_layouts/pptInsertPicture.ashx`.

The parameters to the endpoint are specified by the following table:

Parameter name	Definition
PresentationId	MUST be a valid identifier of a presentation as specified by section 3.1.4.11 .
SlideId	MUST be a valid ST_SlideId ([ISO/IEC-29500-1] section 19.7.13) that specifies the identifier of the presentation slide.
ShapeId	MUST be a valid ST_DrawingElementId ([ISO/IEC-29500-1] section 20.1.10.21) that specifies the identifier of the shape.

[<11> Section 3.1.4:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<12> Section 3.1.4:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<13> Section 3.1.4:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<14> Section 3.1.4.1:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<15> Section 3.1.4.12:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<16> Section 3.1.4.17:](#) Support for this operation was added in Office 2010 Service Pack 1.

[<17> Section 5.1:](#) PowerPoint Web App Technical Preview uses the canary mitigation as follows. On a request to [GetPresentationId](#), a HTTP cookie with a name of X-Key is included in the response. The value of this cookie is then required to be included by subsequent operation calls that may result in a change to a presentation. For any subsequent operation, an additional cookie with the name X-NewKey may also be included which allows the protocol server to inform the protocol client of the new canary value, for example when the canary has expired.

8 Change Tracking

This section identifies changes that were made to the [MS-PWEDPS] protocol document between the March 2011 and June 2011 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
2.2.4.9 ClippingInfo	Added complex type ClippingInfo.	Y	New content added due to protocol revision.
2.2.4.12 EditPresentationChanges	Added new elements to EditPresentationChanges.	Y	New content added due to protocol revision.
2.2.4.13 EditPresentationInfo	Added new element to EditPresentationInfo.	Y	New content added due to protocol revision.
2.2.4.23 PptViewingService.PrintResult	Added complex type PrintResult to support printing.	Y	New content added due to protocol revision.
3.1.4.1 ApplyTheme	Added operation ApplyTheme to support changing theme.	Y	New content added due to protocol revision.
3.1.4.12 InsertClipart	Added operation InsertClipart to support insert pictures.	Y	New content added due to protocol revision.
3.1.4.17 Print	Added operation Print to support printing.	Y	New content added due to protocol revision.

9 Index

A

Abstract data model
 [server](#) 38
[Applicability](#) 14
[ArrayOfContentMasterFe complex type](#) 18
[ArrayOfEditShape complex type](#) 19
[ArrayOfEditSlide complex type](#) 19
[ArrayOfEditSlideChanges complex type](#) 19
[ArrayOfEditSlideInfoFe complex type](#) 19
[ArrayOfMainMasterFe complex type](#) 20
[ArrayOfunsignedInt complex type](#) 20
[ArrayOfViewElement complex type](#) 20
[Attribute groups](#) 36
[Attributes](#) 36

C

[Capability negotiation](#) 14
[Change tracking](#) 128
[char simple type](#) 34
Client
 [overview](#) 37
[ClientActions simple type](#) 34
[ClippingInfo complex type](#) 21
[Complex types](#) 16
 [ArrayOfContentMasterFe](#) 18
 [ArrayOfEditShape](#) 19
 [ArrayOfEditSlide](#) 19
 [ArrayOfEditSlideChanges](#) 19
 [ArrayOfEditSlideInfoFe](#) 19
 [ArrayOfMainMasterFe](#) 20
 [ArrayOfunsignedInt](#) 20
 [ArrayOfViewElement](#) 20
 [ClippingInfo](#) 21
 [ContentMasterFe](#) 21
 [EditCommandResponse](#) 21
 [EditPresentationChanges](#) 22
 [EditPresentationInfo](#) 23
 [EditShape](#) 24
 [EditSlide](#) 25
 [EditSlideChanges](#) 26
 [EditSlideInfoFe](#) 26
 [FontSizeMapping](#) 27
 [Html](#) 27
 [MainMasterFe](#) 28
 [Notes](#) 28
 [Point](#) 28
 [PptViewingService.PrintResult](#) 29
 [Rectangle](#) 29
 [ServiceError](#) 30
 [ServiceResult](#) 30
 [SlideIdList](#) 30
 [SmartArtInfo](#) 31
 [Theme](#) 31
 [ThemeColor](#) 32
 [ViewElement](#) 32
[ContentMasterFe complex type](#) 21

D

Data model - abstract
 [server](#) 38
[duration simple type](#) 35

E

[EditCommandResponse complex type](#) 21
[EditPresentationChanges complex type](#) 22
[EditPresentationInfo complex type](#) 23
[EditShape complex type](#) 24
[EditSlide complex type](#) 25
[EditSlideChanges complex type](#) 26
[EditSlideInfoFe complex type](#) 26
Events
 [local - server](#) 91
 [timer - server](#) 91
Examples
 [overview](#) 92
 [sample protocol interaction](#) 92

F

[Fields - vendor-extensible](#) 14
[FontSizeMapping complex type](#) 27
[Full WSDL](#) 104

G

[Glossary](#) 11
[Groups](#) 36
[guid simple type](#) 35

H

[Html complex type](#) 27

I

[Implementer - security considerations](#) 103
[Index of security parameters](#) 103
[Informative references](#) 12
Initialization
 [server](#) 38
[Introduction](#) 11

L

Local events
 [server](#) 91

M

[MainMasterFe complex type](#) 28
Message processing
 [server](#) 38

- [Messages](#) 15
 - [ArrayOfContentMasterFe complex type](#) 18
 - [ArrayOfEditShape complex type](#) 19
 - [ArrayOfEditSlide complex type](#) 19
 - [ArrayOfEditSlideChanges complex type](#) 19
 - [ArrayOfEditSlideInfoFe complex type](#) 19
 - [ArrayOfMainMasterFe complex type](#) 20
 - [ArrayOfunsignedInt complex type](#) 20
 - [ArrayOfViewElement complex type](#) 20
 - [attribute groups](#) 36
 - [attributes](#) 36
 - [char simple type](#) 34
 - [ClientActions simple type](#) 34
 - [ClippingInfo complex type](#) 21
 - [complex types](#) 16
 - [ContentMasterFe complex type](#) 21
 - [duration simple type](#) 35
 - [EditCommandResponse complex type](#) 21
 - [EditPresentationChanges complex type](#) 22
 - [EditPresentationInfo complex type](#) 23
 - [EditShape complex type](#) 24
 - [EditSlide complex type](#) 25
 - [EditSlideChanges complex type](#) 26
 - [EditSlideInfoFe complex type](#) 26
 - [elements](#) 16
 - [enumerated](#) 15
 - [FontSizeMapping complex type](#) 27
 - [groups](#) 36
 - [guid simple type](#) 35
 - [Html complex type](#) 27
 - [MainMasterFe complex type](#) 28
 - [namespaces](#) 15
 - [Notes complex type](#) 28
 - [Point complex type](#) 28
 - [PptViewingService.PrintResult complex type](#) 29
 - [Rectangle complex type](#) 29
 - [ServiceError complex type](#) 30
 - [ServiceErrorType simple type](#) 35
 - [ServiceResult complex type](#) 30
 - [simple types](#) 33
 - [SlideIdList complex type](#) 30
 - [SmartArtInfo complex type](#) 31
 - [syntax](#) 15
 - [Theme complex type](#) 31
 - [ThemeColor complex type](#) 32
 - [transport](#) 15
 - [ViewElement complex type](#) 32

N

- [Namespaces](#) 15
- [Normative references](#) 11
- [Notes complex type](#) 28

O

- Operations
 - [ApplyTheme](#) 39
 - [ChangePictureStyle](#) 41
 - [ChangeSmartArtColor](#) 43
 - [ChangeSmartArtLayout](#) 45
 - [ChangeSmartArtStyle](#) 47

- [ClearPlaceholder](#) 49
- [DeleteSlide](#) 51
- [DuplicateSlide](#) 53
- [GetEditPresInfo](#) 55
- [GetEditSlide](#) 57
- [GetPresentationId](#) 59
- [InsertClipart](#) 61
- [InsertSlide](#) 63
- [InsertSmartArt](#) 65
- [MoveSlide](#) 67
- [Noop](#) 69
- [Print](#) 71
- [Redo](#) 72
- [ReplaceNotes](#) 74
- [ReplaceText](#) 76
- [ResetSmartArt](#) 78
- [ReverseSmartArt](#) 80
- [SaveAndClose](#) 82
- [ShowHideSlide](#) 84
- [Undo](#) 86
- [UpdateCanary](#) 88
- [WaitCrash](#) 89
- [Overview \(synopsis\)](#) 13

P

- [Parameters - security index](#) 103
- [Point complex type](#) 28
- [PptViewingService.PrintResult complex type](#) 29
- [Preconditions](#) 13
- [Prerequisites](#) 13
- [Product behavior](#) 126

R

- [Rectangle complex type](#) 29
- References
 - [informative](#) 12
 - [normative](#) 11
- [Relationship to other protocols](#) 13

S

- [Sample protocol interaction example](#) 92
- Security
 - [implementer considerations](#) 103
 - [parameter index](#) 103
- Sequencing rules
 - [server](#) 38
- Server
 - [abstract data model](#) 38
 - [ApplyTheme operation](#) 39
 - [ChangePictureStyle operation](#) 41
 - [ChangeSmartArtColor operation](#) 43
 - [ChangeSmartArtLayout operation](#) 45
 - [ChangeSmartArtStyle operation](#) 47
 - [ClearPlaceholder operation](#) 49
 - [DeleteSlide operation](#) 51
 - [details](#) 37
 - [DuplicateSlide operation](#) 53
 - [GetEditPresInfo operation](#) 55
 - [GetEditSlide operation](#) 57

- [GetPresentationId operation](#) 59
- [initialization](#) 38
- [InsertClipart operation](#) 61
- [InsertSlide operation](#) 63
- [InsertSmartArt operation](#) 65
- [local events](#) 91
- [message processing](#) 38
- [MoveSlide operation](#) 67
- [Noop operation](#) 69
- [overview](#) 37
- [Print operation](#) 71
- [Redo operation](#) 72
- [ReplaceNotes operation](#) 74
- [ReplaceText operation](#) 76
- [ResetSmartArt operation](#) 78
- [ReverseSmartArt operation](#) 80
- [SaveAndClose operation](#) 82
- [sequencing rules](#) 38
- [ShowHideSlide operation](#) 84
- [timer events](#) 91
- [timers](#) 38
- [Undo operation](#) 86
- [UpdateCanary operation](#) 88
- [WaitCrash operation](#) 89
- [ServiceError complex type](#) 30
- [ServiceErrorType simple type](#) 35
- [ServiceResult complex type](#) 30
- [Simple types](#) 33
 - [char](#) 34
 - [ClientActions](#) 34
 - [duration](#) 35
 - [guid](#) 35
 - [ServiceErrorType](#) 35
- [SlideIdList complex type](#) 30
- [SmartArtInfo complex type](#) 31
- [Standards assignments](#) 14
- Syntax
 - [messages - overview](#) 15

T

- [Theme complex type](#) 31
- [ThemeColor complex type](#) 32
- Timer events
 - [server](#) 91
- Timers
 - [server](#) 38
- [Tracking changes](#) 128
- [Transport](#) 15
- Types
 - [complex](#) 16
 - [simple](#) 33

V

- [Vendor-extensible fields](#) 14
- [Versioning](#) 14
- [ViewElement complex type](#) 32

W

- [WSDL](#) 104