

[MS-ESP]:

Excel Web Services Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial Availability
6/27/2008	1.0	Major	Revised and edited technical content
10/6/2008	1.01	Minor	Revised and edited technical content
12/12/2008	1.02	Minor	Revised and edited technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	2.0	Minor	Updated the technical content
3/31/2010	2.01	Editorial	Revised and edited the technical content
4/30/2010	2.02	Editorial	Revised and edited the technical content
6/7/2010	2.03	Editorial	Revised and edited the technical content
6/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.04	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2012	3.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.0.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	3.0.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.0.1	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
11/18/2013	3.1	Minor	Clarified the meaning of the technical content.
2/10/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.2	Minor	Clarified the meaning of the technical content.
7/31/2014	3.3	Minor	Clarified the meaning of the technical content.
10/30/2014	3.3	None	No changes to the meaning, language, or formatting of the technical content.
2/26/2016	4.0	Major	Significantly changed the technical content.
7/15/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
6/20/2017	5.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	13
1.1	Glossary	13
1.2	References	17
1.2.1	Normative References	17
1.2.2	Informative References	19
1.3	Overview	19
1.3.1	Published Items and Results Returned by this Protocol	19
1.4	Relationship to Other Protocols	20
1.5	Prerequisites/Preconditions	20
1.6	Applicability Statement	20
1.7	Versioning and Capability Negotiation	20
1.8	Vendor-Extensible Fields	20
1.9	Standards Assignments	20
2	Messages	21
2.1	Transport	21
2.2	Common Message Syntax	21
2.2.1	Namespaces	21
2.2.2	Messages	21
2.2.3	Elements	22
2.2.4	Complex Types	22
2.2.4.1	ArrayOfAnyType	22
2.2.4.2	ArrayOfStatus	22
2.2.4.3	ArrayOfString	23
2.2.4.4	RangeCoordinates	23
2.2.4.5	Status	23
2.2.4.6	TypeExporter	24
2.2.5	Simple Types	24
2.2.5.1	CellError	25
2.2.5.2	RangeReference	25
2.2.5.3	SaveOptions	26
2.2.5.4	SheetName	26
2.2.5.5	WorkbookType	27
2.2.6	Attributes	27
2.2.7	Groups	27
2.2.8	Attribute Groups	27
2.2.9	Common Data Structures	28
2.2.9.1	Faults	28
2.2.9.2	Common List of Subcodes	37
3	Protocol Details	40
3.1	Server Details	41
3.1.1	Abstract Data Model	41
3.1.2	Timers	41
3.1.2.1	Session Timeout	41
3.1.2.2	Request Timeout	41
3.1.3	Initialization	41
3.1.4	Message Processing Events and Sequencing Rules	41
3.1.4.1	Calculate	43
3.1.4.1.1	Messages	44
3.1.4.1.1.1	CalculateSoapIn	44
3.1.4.1.1.2	CalculateSoapOut	44
3.1.4.1.2	Elements	44
3.1.4.1.2.1	Calculate	44
3.1.4.1.2.2	CalculateResponse	45

3.1.4.1.3	Complex Types	45
3.1.4.1.4	Simple Types	45
3.1.4.1.5	Attributes	45
3.1.4.1.6	Groups	45
3.1.4.1.7	Attribute Groups	45
3.1.4.2	CalculateA1	45
3.1.4.2.1	Messages	46
3.1.4.2.1.1	CalculateA1SoapIn	47
3.1.4.2.1.2	CalculateA1SoapOut	47
3.1.4.2.2	Elements	47
3.1.4.2.2.1	CalculateA1	47
3.1.4.2.2.2	CalculateA1Response	47
3.1.4.2.3	Complex Types	48
3.1.4.2.4	Simple Types	48
3.1.4.2.5	Attributes	48
3.1.4.2.6	Groups	48
3.1.4.2.7	Attribute Groups	48
3.1.4.3	CalculateWorkbook	48
3.1.4.3.1	Messages	49
3.1.4.3.1.1	CalculateWorkbookSoapIn	49
3.1.4.3.1.2	CalculateWorkbookSoapOut	49
3.1.4.3.2	Elements	49
3.1.4.3.2.1	CalculateWorkbook	50
3.1.4.3.2.2	CalculateWorkbookResponse	50
3.1.4.3.3	Complex Types	50
3.1.4.3.4	Simple Types	50
3.1.4.3.4.1	CalculateType	50
3.1.4.3.5	Attributes	51
3.1.4.3.6	Groups	51
3.1.4.3.7	Attribute Groups	51
3.1.4.4	CancelRequest	51
3.1.4.4.1	Messages	51
3.1.4.4.1.1	CancelRequestSoapIn	52
3.1.4.4.1.2	CancelRequestSoapOut	52
3.1.4.4.2	Elements	52
3.1.4.4.2.1	CancelRequest	52
3.1.4.4.2.2	CancelRequestResponse	53
3.1.4.4.3	Complex Types	53
3.1.4.4.4	Simple Types	53
3.1.4.4.5	Attributes	53
3.1.4.4.6	Groups	53
3.1.4.4.7	Attribute Groups	53
3.1.4.5	CloseWorkbook	53
3.1.4.5.1	Messages	53
3.1.4.5.1.1	CloseWorkbookSoapIn	54
3.1.4.5.1.2	CloseWorkbookSoapOut	54
3.1.4.5.2	Elements	54
3.1.4.5.2.1	CloseWorkbook	54
3.1.4.5.2.2	CloseWorkbookResponse	55
3.1.4.5.3	Complex Types	55
3.1.4.5.4	Simple Types	55
3.1.4.5.5	Attributes	55
3.1.4.5.6	Groups	55
3.1.4.5.7	Attribute Groups	55
3.1.4.6	EnsureWorkbookModel	55
3.1.4.6.1	Messages	55
3.1.4.6.1.1	EnsureWorkbookModelSoapIn	56
3.1.4.6.1.2	EnsureWorkbookModelSoapOut	56

3.1.4.6.2	Elements	56
3.1.4.6.2.1	EnsureWorkbookModel	56
3.1.4.6.2.2	EnsureWorkbookModelResponse	57
3.1.4.6.3	Complex Types	57
3.1.4.6.3.1	WorkbookModelInfo	57
3.1.4.6.4	Simple Types	58
3.1.4.6.5	Attributes	58
3.1.4.6.6	Groups.....	58
3.1.4.6.7	Attribute Groups.....	58
3.1.4.7	GetApiVersion.....	58
3.1.4.7.1	Messages	58
3.1.4.7.1.1	GetApiVersionSoapIn.....	58
3.1.4.7.1.2	GetApiVersionSoapOut	59
3.1.4.7.2	Elements	59
3.1.4.7.2.1	GetApiVersion	59
3.1.4.7.2.2	GetApiVersionResponse	59
3.1.4.7.3	Complex Types	59
3.1.4.7.4	Simple Types	59
3.1.4.7.5	Attributes	59
3.1.4.7.6	Groups.....	60
3.1.4.7.7	Attribute Groups.....	60
3.1.4.8	GetCell	60
3.1.4.8.1	Messages	61
3.1.4.8.1.1	GetCellSoapIn	61
3.1.4.8.1.2	GetCellSoapOut	61
3.1.4.8.2	Elements.....	61
3.1.4.8.2.1	GetCell	61
3.1.4.8.2.2	GetCellResponse	62
3.1.4.8.3	Complex Types	62
3.1.4.8.4	Simple Types	62
3.1.4.8.5	Attributes	62
3.1.4.8.6	Groups.....	62
3.1.4.8.7	Attribute Groups.....	62
3.1.4.9	GetCellA1.....	62
3.1.4.9.1	Messages	63
3.1.4.9.1.1	GetCellA1SoapIn.....	64
3.1.4.9.1.2	GetCellA1SoapOut	64
3.1.4.9.2	Elements.....	64
3.1.4.9.2.1	GetCellA1	64
3.1.4.9.2.2	GetCellA1Response	65
3.1.4.9.3	Complex Types	65
3.1.4.9.4	Simple Types	65
3.1.4.9.5	Attributes	65
3.1.4.9.6	Groups.....	65
3.1.4.9.7	Attribute Groups.....	65
3.1.4.10	GetChartImageUrl.....	65
3.1.4.10.1	Messages	66
3.1.4.10.1.1	GetChartImageUrlSoapIn.....	66
3.1.4.10.1.2	GetChartImageUrlSoapOut.....	66
3.1.4.10.2	Elements.....	66
3.1.4.10.2.1	GetChartImageUrl.....	66
3.1.4.10.2.2	GetChartImageUrlResponse	67
3.1.4.10.3	Complex Types	67
3.1.4.10.3.1	Size	67
3.1.4.10.4	Simple Types	68
3.1.4.10.5	Attributes	68
3.1.4.10.6	Groups.....	68
3.1.4.10.7	Attribute Groups.....	68

3.1.4.11	GetPublishedItemNames	68
3.1.4.11.1	Messages	68
3.1.4.11.1.1	GetPublishedItemNamesSoapIn	69
3.1.4.11.1.2	GetPublishedItemNamesSoapOut	69
3.1.4.11.2	Elements	69
3.1.4.11.2.1	GetPublishedItemNames	69
3.1.4.11.2.2	GetPublishedItemNamesResponse	69
3.1.4.11.3	Complex Types	70
3.1.4.11.3.1	ArrayOfWorkbookItem	70
3.1.4.11.3.2	WorkbookItem	70
3.1.4.11.4	Simple Types	70
3.1.4.11.4.1	ItemType	71
3.1.4.11.5	Attributes	71
3.1.4.11.6	Groups	71
3.1.4.11.7	Attribute Groups	71
3.1.4.12	GetRange	71
3.1.4.12.1	Messages	72
3.1.4.12.1.1	GetRangeSoapIn	73
3.1.4.12.1.2	GetRangeSoapOut	73
3.1.4.12.2	Elements	73
3.1.4.12.2.1	GetRange	73
3.1.4.12.2.2	GetRangeResponse	74
3.1.4.12.3	Complex Types	74
3.1.4.12.4	Simple Types	74
3.1.4.12.5	Attributes	74
3.1.4.12.6	Groups	74
3.1.4.12.7	Attribute Groups	74
3.1.4.13	GetRangeA1	74
3.1.4.13.1	Messages	76
3.1.4.13.1.1	GetRangeA1SoapIn	76
3.1.4.13.1.2	GetRangeA1SoapOut	76
3.1.4.13.2	Elements	76
3.1.4.13.2.1	GetRangeA1	76
3.1.4.13.2.2	GetRangeA1Response	77
3.1.4.13.3	Complex Types	77
3.1.4.13.4	Simple Types	77
3.1.4.13.5	Attributes	77
3.1.4.13.6	Groups	77
3.1.4.13.7	Attribute Groups	77
3.1.4.14	GetSessionInformation	77
3.1.4.14.1	Messages	78
3.1.4.14.1.1	GetSessionInformationSoapIn	78
3.1.4.14.1.2	GetSessionInformationSoapOut	78
3.1.4.14.2	Elements	78
3.1.4.14.2.1	GetSessionInformation	79
3.1.4.14.2.2	GetSessionInformationResponse	79
3.1.4.14.3	Complex Types	79
3.1.4.14.4	Simple Types	79
3.1.4.14.5	Attributes	79
3.1.4.14.6	Groups	80
3.1.4.14.7	Attribute Groups	80
3.1.4.15	GetSheetNames	80
3.1.4.15.1	Messages	80
3.1.4.15.1.1	GetSheetNamesSoapIn	80
3.1.4.15.1.2	GetSheetNamesSoapOut	81
3.1.4.15.2	Elements	81
3.1.4.15.2.1	GetSheetNames	81
3.1.4.15.2.2	GetSheetNamesResponse	81

3.1.4.15.3	Complex Types	81
3.1.4.15.3.1	ArrayOfSheetInfo	82
3.1.4.15.3.2	SheetInfo	82
3.1.4.15.4	Simple Types	82
3.1.4.15.4.1	SheetType	83
3.1.4.15.4.2	SheetVisibility	83
3.1.4.15.5	Attributes	83
3.1.4.15.6	Groups	83
3.1.4.15.7	Attribute Groups	83
3.1.4.16	GetWorkbook	84
3.1.4.16.1	Messages	84
3.1.4.16.1.1	GetWorkbookSoapIn	84
3.1.4.16.1.2	GetWorkbookSoapOut	85
3.1.4.16.2	Elements	85
3.1.4.16.2.1	GetWorkbook	85
3.1.4.16.2.2	GetWorkbookResponse	85
3.1.4.16.3	Complex Types	86
3.1.4.16.4	Simple Types	86
3.1.4.16.5	Attributes	86
3.1.4.16.6	Groups	86
3.1.4.16.7	Attribute Groups	86
3.1.4.17	GetWorkbookConnections	86
3.1.4.17.1	Messages	86
3.1.4.17.1.1	GetWorkbookConnectionsSoapIn	87
3.1.4.17.1.2	GetWorkbookConnectionsSoapOut	87
3.1.4.17.2	Elements	87
3.1.4.17.2.1	GetWorkbookConnections	87
3.1.4.17.2.2	GetWorkbookConnectionsResponse	87
3.1.4.17.3	Complex Types	88
3.1.4.17.4	Simple Types	88
3.1.4.17.5	Attributes	88
3.1.4.17.6	Groups	88
3.1.4.17.7	Attribute Groups	88
3.1.4.18	NewWorkbook	88
3.1.4.18.1	Messages	89
3.1.4.18.1.1	NewWorkbookSoapIn	89
3.1.4.18.1.2	NewWorkbookSoapOut	89
3.1.4.18.2	Elements	89
3.1.4.18.2.1	NewWorkbook	89
3.1.4.18.2.2	NewWorkbookResponse	90
3.1.4.18.3	Complex Types	90
3.1.4.18.4	Simple Types	90
3.1.4.18.5	Attributes	90
3.1.4.18.6	Groups	90
3.1.4.18.7	Attribute Groups	90
3.1.4.19	OpenWorkbook	90
3.1.4.19.1	Messages	92
3.1.4.19.1.1	OpenWorkbookSoapIn	92
3.1.4.19.1.2	OpenWorkbookSoapOut	92
3.1.4.19.2	Elements	92
3.1.4.19.2.1	OpenWorkbook	92
3.1.4.19.2.2	OpenWorkbookResponse	93
3.1.4.19.3	Complex Types	93
3.1.4.19.4	Simple Types	93
3.1.4.19.5	Attributes	93
3.1.4.19.6	Groups	93
3.1.4.19.7	Attribute Groups	93
3.1.4.20	OpenWorkbookEx	94

3.1.4.20.1	Messages	94
3.1.4.20.1.1	OpenWorkbookExSoapIn.....	94
3.1.4.20.1.2	OpenWorkbookExSoapOut	94
3.1.4.20.2	Elements.....	94
3.1.4.20.2.1	OpenWorkbookEx	95
3.1.4.20.2.2	OpenWorkbookExResponse	95
3.1.4.20.3	Complex Types	95
3.1.4.20.4	Simple Types	95
3.1.4.20.5	Attributes	95
3.1.4.20.6	Groups.....	95
3.1.4.20.7	Attribute Groups.....	96
3.1.4.21	OpenWorkbookForEditing	96
3.1.4.21.1	Messages	97
3.1.4.21.1.1	OpenWorkbookForEditingSoapIn	97
3.1.4.21.1.2	OpenWorkbookForEditingSoapOut.....	97
3.1.4.21.2	Elements.....	97
3.1.4.21.2.1	OpenWorkbookForEditing.....	98
3.1.4.21.2.2	OpenWorkbookForEditingResponse	98
3.1.4.21.3	Complex Types	98
3.1.4.21.4	Simple Types	99
3.1.4.21.5	Attributes	99
3.1.4.21.6	Groups.....	99
3.1.4.21.7	Attribute Groups.....	99
3.1.4.22	Refresh.....	99
3.1.4.22.1	Messages	99
3.1.4.22.1.1	RefreshSoapIn.....	99
3.1.4.22.1.2	RefreshSoapOut.....	100
3.1.4.22.2	Elements.....	100
3.1.4.22.2.1	Refresh.....	100
3.1.4.22.2.2	RefreshResponse	100
3.1.4.22.3	Complex Types	101
3.1.4.22.4	Simple Types	101
3.1.4.22.5	Attributes	101
3.1.4.22.6	Groups.....	101
3.1.4.22.7	Attribute Groups.....	101
3.1.4.23	RefreshDetached.....	101
3.1.4.23.1	Messages	101
3.1.4.23.1.1	RefreshDetachedSoapIn.....	101
3.1.4.23.1.2	RefreshDetachedSoapOut	102
3.1.4.23.2	Elements.....	102
3.1.4.23.2.1	RefreshDetached	102
3.1.4.23.2.2	RefreshDetachedResponse	102
3.1.4.23.3	Complex Types	103
3.1.4.23.3.1	ArrayOfConnectionDetails	103
3.1.4.23.3.2	ArrayOfDetailedRefreshResult.....	103
3.1.4.23.3.3	ConnectionDetails	103
3.1.4.23.3.4	DetailedRefreshResult	104
3.1.4.23.4	Simple Types	104
3.1.4.23.5	Attributes	104
3.1.4.23.6	Groups.....	104
3.1.4.23.7	Attribute Groups.....	104
3.1.4.24	RefreshEx	104
3.1.4.24.1	Messages	104
3.1.4.24.1.1	RefreshExSoapIn	105
3.1.4.24.1.2	RefreshExSoapOut	105
3.1.4.24.2	Elements.....	105
3.1.4.24.2.1	RefreshEx	105
3.1.4.24.2.2	RefreshExResponse	106

3.1.4.24.3	Complex Types	106
3.1.4.24.4	Simple Types	106
3.1.4.24.5	Attributes	106
3.1.4.24.6	Groups.....	106
3.1.4.24.7	Attribute Groups.....	106
3.1.4.25	SaveWorkbook	106
3.1.4.25.1	Messages	107
3.1.4.25.1.1	SaveWorkbookSoapIn	107
3.1.4.25.1.2	SaveWorkbookSoapOut	107
3.1.4.25.2	Elements.....	107
3.1.4.25.2.1	SaveWorkbook	108
3.1.4.25.2.2	SaveWorkbookResponse	108
3.1.4.25.3	Complex Types	108
3.1.4.25.4	Simple Types	108
3.1.4.25.5	Attributes	108
3.1.4.25.6	Groups.....	108
3.1.4.25.7	Attribute Groups.....	108
3.1.4.26	SaveWorkbookCopy	109
3.1.4.26.1	Messages	109
3.1.4.26.1.1	SaveWorkbookCopySoapIn	110
3.1.4.26.1.2	SaveWorkbookCopySoapOut	110
3.1.4.26.2	Elements.....	110
3.1.4.26.2.1	SaveWorkbookCopy	110
3.1.4.26.2.2	SaveWorkbookCopyResponse	111
3.1.4.26.3	Complex Types	111
3.1.4.26.4	Simple Types	111
3.1.4.26.5	Attributes	111
3.1.4.26.6	Groups.....	111
3.1.4.26.7	Attribute Groups.....	111
3.1.4.27	SaveWorkbookCopyDetached	111
3.1.4.27.1	Messages	112
3.1.4.27.1.1	SaveWorkbookCopyDetachedSoapIn	112
3.1.4.27.1.2	SaveWorkbookCopyDetachedSoapOut	112
3.1.4.27.2	Elements.....	112
3.1.4.27.2.1	SaveWorkbookCopyDetached	112
3.1.4.27.2.2	SaveWorkbookCopyDetachedResponse	113
3.1.4.27.3	Complex Types	113
3.1.4.27.4	Simple Types	113
3.1.4.27.5	Attributes	113
3.1.4.27.6	Groups.....	113
3.1.4.27.7	Attribute Groups.....	113
3.1.4.28	SetCalculationOptions.....	113
3.1.4.28.1	Messages	114
3.1.4.28.1.1	SetCalculationOptionsSoapIn.....	114
3.1.4.28.1.2	SetCalculationOptionsSoapOut.....	114
3.1.4.28.2	Elements.....	114
3.1.4.28.2.1	SetCalculationOptions.....	115
3.1.4.28.2.2	SetCalculationOptionsResponse	115
3.1.4.28.3	Complex Types	115
3.1.4.28.4	Simple Types	115
3.1.4.28.4.1	WorkbookCalculation	115
3.1.4.28.5	Attributes	116
3.1.4.28.6	Groups.....	116
3.1.4.28.7	Attribute Groups.....	116
3.1.4.29	SetCell.....	116
3.1.4.29.1	Messages	117
3.1.4.29.1.1	SetCellSoapIn.....	117
3.1.4.29.1.2	SetCellSoapOut	117

3.1.4.29.2	Elements	117
3.1.4.29.2.1	SetCell	118
3.1.4.29.2.2	SetCellResponse	118
3.1.4.29.3	Complex Types	118
3.1.4.29.4	Simple Types	118
3.1.4.29.5	Attributes	119
3.1.4.29.6	Groups.....	119
3.1.4.29.7	Attribute Groups.....	119
3.1.4.30	SetCellA1	119
3.1.4.30.1	Messages	120
3.1.4.30.1.1	SetCellA1SoapIn	120
3.1.4.30.1.2	SetCellA1SoapOut.....	120
3.1.4.30.2	Elements	120
3.1.4.30.2.1	SetCellA1	121
3.1.4.30.2.2	SetCellA1Response	121
3.1.4.30.3	Complex Types	121
3.1.4.30.4	Simple Types	121
3.1.4.30.5	Attributes	121
3.1.4.30.6	Groups.....	121
3.1.4.30.7	Attribute Groups.....	121
3.1.4.31	SetParameters.....	122
3.1.4.31.1	Messages	123
3.1.4.31.1.1	SetParametersSoapIn.....	123
3.1.4.31.1.2	SetParametersSoapOut.....	123
3.1.4.31.2	Elements	123
3.1.4.31.2.1	SetParameters.....	123
3.1.4.31.2.2	SetParametersResponse	124
3.1.4.31.3	Complex Types	124
3.1.4.31.3.1	ArrayOfParameterInfo	124
3.1.4.31.3.2	ParameterInfo	124
3.1.4.31.4	Simple Types	125
3.1.4.31.5	Attributes	125
3.1.4.31.6	Groups.....	125
3.1.4.31.7	Attribute Groups.....	125
3.1.4.32	SetRange	125
3.1.4.32.1	Messages	126
3.1.4.32.1.1	SetRangeSoapIn	126
3.1.4.32.1.2	SetRangeSoapOut.....	126
3.1.4.32.2	Elements	126
3.1.4.32.2.1	SetRange.....	127
3.1.4.32.2.2	SetRangeResponse	127
3.1.4.32.3	Complex Types	127
3.1.4.32.4	Simple Types	127
3.1.4.32.5	Attributes	128
3.1.4.32.6	Groups.....	128
3.1.4.32.7	Attribute Groups.....	128
3.1.4.33	SetRangeA1	128
3.1.4.33.1	Messages	128
3.1.4.33.1.1	SetRangeA1SoapIn	129
3.1.4.33.1.2	SetRangeA1SoapOut	129
3.1.4.33.2	Elements	129
3.1.4.33.2.1	SetRangeA1	129
3.1.4.33.2.2	SetRangeA1Response.....	130
3.1.4.33.3	Complex Types	130
3.1.4.33.4	Simple Types	130
3.1.4.33.5	Attributes	130
3.1.4.33.6	Groups.....	130
3.1.4.33.7	Attribute Groups.....	130

3.1.5	Timer Events.....	130
3.1.6	Other Local Events.....	130
4	Protocol Examples	131
4.1	Getting the Value of a Worksheet Cell.....	131
4.2	Refreshing Data and Retrieving a Workbook	134
5	Security	147
5.1	Security Considerations for Implementers	147
5.2	Index of Security Parameters	147
6	Appendix A: Full WSDL	148
7	Appendix B: Product Behavior	175
8	Change Tracking.....	177
9	Index.....	178

1 Introduction

The Excel Web Services Protocol is a Web service protocol that is used to read from and perform basic operations on workbooks hosted on a protocol server.

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

1.1 Glossary

This document uses the following terms:

active sheet: The sheet that is currently selected.

array formula: A formula that performs multiple calculations on one or more sets of values, and then returns either a single result or multiple results. Array formulas are enclosed in braces ({}), and are entered by pressing CTRL+SHIFT+ENTER.

AutoFilter: A mechanism that can be used to filter tabular data based on user-defined criteria such as values, strings, and formatting.

automatic calculation mode: A **calculation mode** in which formulas in a worksheet are recalculated automatically whenever the data changes. See also **manual calculation mode**.

calculate: The process by which computations in a workbook are performed.

calculation mode: A setting that determines whether the formulas in a worksheet are recalculated automatically or manually. See also **automatic calculation mode** and **manual calculation mode**.

cell: A box that is formed by the intersection of a **row** and a **column** in a worksheet or a table. A cell can contain numbers, strings, and formulas, and various formats can be applied to that data.

cell error value: Any of a number of special values that are returned as a result of an unsuccessful formula calculation.

chart sheet: A single logical container that is used to create and store charts in a workbook.

checked out: A publishing level that indicates that a document has been created and locked for exclusive editing by a user in a version control system.

column: A single set of data that is displayed vertically in a worksheet or a table.

confirmation: A message that asks a user to verify an action before the user performs it.

connected data: Data that is stored in the same workbook from which it is being referenced, or data that is stored in a database repository.

cube: A set of data that is organized and summarized into a multidimensional structure that is defined by a set of dimensions and measures.

data connection: A collection of information, such as the type and location, that defines how to connect to an external data source, such as a database, **web service**, SharePoint list, or XML file.

data culture: The language that is used to specify number formatting for data.

data provider: A known data source that is specific to a target type and that provides data to a collector type.

data table: A range of cells that is designated to perform what-if analysis for formulas, based on various input values.

default filter value: The filter value that is used if no filter is specified.

dimension: A categorization of data in rows or **columns** in an Excel worksheet.

document library: A type of list that is a container for documents and folders.

edit session: A **session (2)** that is based on a workbook file and during which changes by one or more users are persisted to that file.

farm: A group of computers that work together as a single system to help ensure that applications and resources are available. Also referred to as server farm.

filter: A mechanism by which a set of data is scoped to display only those entries that meet specified logical criteria.

formula: A logical equation or function that produces a result in a spreadsheet application.

grow: The process of increasing the number of rows or **columns** in a PivotTable report.

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

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

Information Rights Management (IRM): A technology that provides persistent protection to digital data by using encryption, certificates, and authentication. Authorized recipients or users acquire a license to gain access to the protected files according to the rights or business rules that are set by the content owner.

item: A unit of content that can be indexed and searched by a search application.

level: A relative position in a hierarchy of data. A level is frequently used when describing how to navigate a hierarchy in an Online Analytical Processing (OLAP) database or a PivotTable report.

license key: An array of bytes that enables access to a control according to the usage policies for that control.

list: An organization of a region of cells into a tabular structure in a workbook.

locked: The condition of a cell, worksheet, or other object that restricts edits or modifications to it by users.

manual calculation mode: A **calculation mode** in which the formulas in a workbook are calculated only when specified by a user. See also **automatic calculation mode**.

manual filter: A filter that lets users select a set of individual data items or dimension hierarchies by using a user interface that displays levels from the hierarchies.

merged cell: A single cell that is created by combining two or more adjacent cells.

named object: A **list**, PivotTable report, chart, or range that can be referenced by name. A sheet is not a type of named object.

named range: See defined name.

new workbook session: A **session (2)** that is not based on an existing workbook file.

non-contiguous range: A selected **range** that includes non-adjacent cells.

number format: A property of a cell or other type of object that determines how numerical data is displayed or interpreted. For example, a currency number format affixes the proper currency symbol to the number.

Online Analytical Processing (OLAP): A technology that uses multidimensional structures to provide access to data for analysis. The source data for OLAP is stored in data warehouses in a relational database. See also **cube**.

Open Item permission: An authorization that enables users to retrieve an entire file.

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

PivotTable: An interactive table that summarizes large amounts of data from various sources by using format and calculation methods. Row and column headings can be rotated to view different summaries of the source data, filter the data, or display detail data for specific areas.

PivotTable member: An individual data item within a specific dimension as it is displayed in a PivotTable report. For example, a member in the Geography dimension might be North America.

PivotTable page field: A filter in a PivotTable report that applies to all of the data in the report, including any hierarchies.

published: A condition of portions of a workbook that are marked as being available to the user when that workbook is processed by a protocol server.

published item: A specific named object that is in a published workbook.

published items snapshot: A snapshot that contains only the published items or published sheets from a workbook. If no published items are defined, a snapshot contains an entire workbook.

published sheet: A **sheet (2)** that is published.

query: A formalized instruction to a data source to either extract data or perform a specified action. A query can be in the form of a query expression, a method-based query, or a combination of the two. The data source can be in different forms, such as a relational database, XML document, or in-memory object. See also search query.

range: An addressable region that is in a workbook. A range typically consists of zero or more cells and represents a single, contiguous rectangle of cells on a single sheet.

refresh: A process that retrieves values from a data source and populates a workbook with those values.

row: A single set of data that is displayed horizontally in a worksheet or a table.

session: (1) A unidirectional communication channel for a stream of messages that are addressed to one or more destinations. A destination is specified by a resource URL, an identity URL, and a device URL. More than one session can be multiplexed over a single connection.

(2) A representation of application data in system memory. It is used to maintain state for application data that is being manipulated or monitored on a protocol server by a user.

session identifier: A key that enables an application to make reference to a session.

sheet: (1) A part of an Excel workbook. There are four types of sheets: **worksheet**, macro sheet, dialog sheet, and **chart sheet**. Multiple sheets are stored together within a workbook.

(2) A worksheet. The term sheet frequently refers to a worksheet because worksheets are the most common type of sheet.

site: A group of related pages and data within a SharePoint site collection. The structure and content of a site is based on a site definition. Also referred to as SharePoint site and web site.

slicer: A mechanism that is used to filter data in one or more PivotTable reports or cube functions.

snapshot: A copy of a workbook that contains only values and formatting. It does not contain any formulas or data connections.

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

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

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

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

table: A **list** that is defined in a workbook.

trusted location: A directory from which files can be opened without being checked by the security features of a Microsoft Office application.

UI culture: The language that is used to display strings and graphical elements in a user interface.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

Universal Naming Convention (UNC): A string format that specifies the location of a resource. For more information, see [\[MS-DTYP\]](#) section 2.2.57.

user-defined function (UDF): A function that is coded in a Microsoft Visual Basic for Applications (VBA) module, macro sheet, add-in, or Excel Linked Library (XLL). A UDF can be used in formulas to return values to a worksheet, similar to built-in functions.

view session: A **session (2)** that is based on a workbook file and during which the file can be viewed by one or more users. Any changes to the file, such as sorting or filtering data, are not persisted to that file.

view-only mode: A display mode in which data in a workbook can only be viewed, sorted, and filtered by users. Users cannot change any of the data in the workbook and any changes to the display of the data are not persisted to the workbook file.

visible: A condition of an object that allows it to be displayed in rendered output.

web service: A unit of application logic that provides data and services to other applications and can be called by using standard Internet transport protocols such as **HTTP**, Simple Mail Transfer Protocol (SMTP), or File Transfer Protocol (FTP). Web services can perform functions that range from simple requests to complicated business processes.

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

workbook: A container for a collection of **sheets (1)**.

workbook file: A file that contains a byte stream representation of a workbook.

workbook parameter: A single cell that is designated to receive input from users.

worksheet: A single logical container for a set of tabular data and other objects in a workbook.

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

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

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

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

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

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

1.2 References

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

1.2.1 Normative References

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

[ISO/IEC29500-1:2011] ISO/IEC, "Information Technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference",

ISO/IEC 29500-1:2011, 2011,
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=59575

[ISO/IEC29500-1:2012] ISO/IEC, "Information Technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference", ISO/IEC 29500-1:2012,
http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=61750

[MS-ODBCSTR] Microsoft Corporation, "[ODBC Connection String Structure](#)".

[MS-ODCFF] Microsoft Corporation, "[Office Data Connection File Format](#)".

[MS-OFFMACRO2] Microsoft Corporation, "[Office Macro-Enabled File Format Version 2](#)".

[MS-OLEDBSTR] Microsoft Corporation, "[OLEDB Connection String Structure](#)".

[MS-XLSB] Microsoft Corporation, "[Excel \(.xlsb\) Binary File Format](#)".

[ODF] OASIS, "Open Document Format for Office Applications (OpenDocument) v1.1", 1 February 2007, <http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1-html/OpenDocument-v1.1.html>

[RFC1766] Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, March 1995,
<http://www.ietf.org/rfc/rfc1766.txt>

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

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

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

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, <http://www.rfc-editor.org/rfc/rfc3986.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

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

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

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

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

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

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

1.2.2 Informative References

[MSDN-ESO] Microsoft Corporation, "Excel Services Overview", <http://msdn.microsoft.com/en-us/library/ms546696.aspx>

[MSDN-UFES] Microsoft Corporation, "Unsupported Features in Excel Services", <http://msdn.microsoft.com/en-us/library/ms496823.aspx>

[MSDN-WDCAEWS] Microsoft Corporation, "Walkthrough: Developing a Custom Application Using Excel Web Services", <http://msdn.microsoft.com/en-us/library/ms519100.aspx>

[MSFT-ACRR] Microsoft Corporation, "Overview of formulas", <https://support.office.com/en-US/article/overview-of-formulas-7abfda78-eff3-4cc6-b4a7-6350d512d2dc>

1.3 Overview

This protocol is a Web service based protocol that is used to access **workbooks** controlled by a protocol server. The protocol specified in this document is stateful and relies on **sessions (2)** to maintain state. Workbooks hosted on a protocol server are used to present, store, manipulate, and **calculate** tabular data. For an overview of the specific implementation of a workbook server system that uses this protocol, see [\[MSDN-ESO\]](#).

The protocol provides the following sets of functionality that can be performed on workbooks:

- Open existing workbooks for viewing and editing and close open workbooks.
- Create new workbooks.
- Save workbooks.
- Retrieve the full contents of the current open copy.
- Recalculation.
- **Refresh connected data.**
- Set specific values into a **worksheet cell** or **range**.
- Get specific values from a worksheet cell or range.
- Cancel a request.
- Get the names of all the **published items** in a workbook.
- Get a **URL** to an image of a chart in a workbook.
- Get information about a session.
- Get the version of the protocol server.

1.3.1 Published Items and Results Returned by this Protocol

A protocol server can have functionality where only portions of the workbook are available for use by protocol clients. The parts of the workbook that are accessible from the protocol server are said to be **published**. Some of the operations in this protocol could fail if the workbook item being exposed or changed is not published and the user does not have sufficient permissions to view items that have not been published.

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:

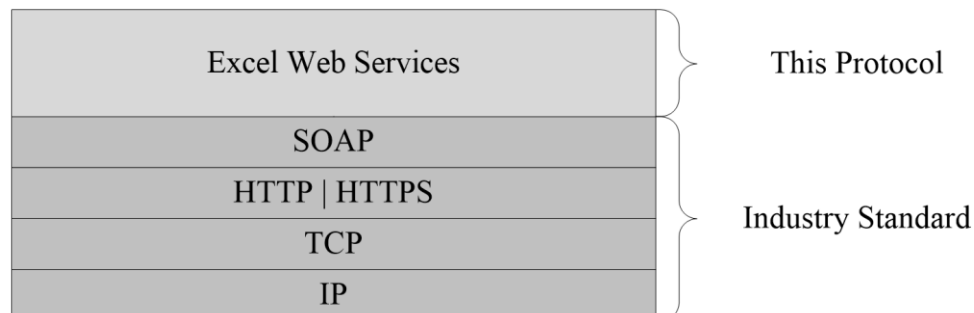


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

This protocol operates against a **site** that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending "_vti_bin/ExcelService.asmx" to the URL of the site, for example http://www.contoso.com/Repository/_vti_bin/ExcelService.asmx.

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

1.6 Applicability Statement

This protocol can be used at any time to access a simple interface that supports basic operations that leverage server side calculation of workbooks is needed. It is appropriate to use this as a public-facing interface to workbooks hosted on a server.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported transports:** This protocol can be implemented by using transports that support sending SOAP messages, as described in section [2.1](#).
- **Protocol versions:** This protocol is not versioned.
- **Capability negotiation:** This protocol does not support capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

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

2.1 Transport

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

Protocol messages **MUST** be formatted as specified either in [\[SOAP1.1\]](#) section 4 or in [\[SOAP1.2/1\]](#) section 5. Protocol server faults **MUST** be returned either by using HTTP Status Codes as specified in [\[RFC2616\]](#) section 10, or by using **SOAP faults** as specified either in [SOAP1.1], section 4.4 or in [SOAP1.2/1] section 5.4. The version of the SOAP fault returned **MUST** correspond to the version of SOAP used for the request **WSDL message**.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and **WSDL**, as specified 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
http	http://schemas.xmlsoap.org/wsdl/http/	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/office/excel/server/webservices	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

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

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
ArrayOfAnyType	An array of anyType elements defined as follows.
ArrayOfStatus	An array of status information.
ArrayOfString	An array of string elements.
RangeCoordinates	Specifies a range on a workbook.
Status	The status information returned by the protocol server defined as follows.
TypeExporter	MUST NOT be used.

2.2.4.1 ArrayOfAnyType

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

An array of **anyType** elements defined as follows.

```
<xs:complexType name="ArrayOfAnyType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="anyType" nillable="true"/>
  </xs:sequence>
</xs:complexType>
```

anyType: An **anyType** value.

2.2.4.2 ArrayOfStatus

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

An array of status information.

- The protocol server MUST use the **Status** complex type to return informative status messages to the protocol client for operations that succeed.
- The protocol client MUST NOT perform any logic based on the specific values in the **Name** or **Message** elements returned in the **Status** type.

This complex type is defined as follows.

```
<xs:complexType name="ArrayOfStatus" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Status" nillable="true"
      type="tns:Status"/>
  </xs:sequence>
```

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

Status: The status information returned by the protocol server.

2.2.4.3 ArrayOfString

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

An array of string elements.

```
<xs:complexType name="ArrayOfString" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
      type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

string: A string value.

2.2.4.4 RangeCoordinates

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

Specifies a **range** on a workbook.

- **RangeCoordinates** MUST be considered empty if **Row**, **Column**, **Height**, and **Width** are all 0. Otherwise, **RangeCoordinates** MUST NOT be considered empty.
- If **Height** is 0, then **Row**, **Column**, and **Width** MUST be 0.
- If **Width** is 0, then **Row**, **Column**, and **Height** MUST be 0.

This complex type is defined as follows.

```
<xs:complexType name="RangeCoordinates" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Row" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Column" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

Row: The **row** index. MUST be greater than or equal to 0 and less than or equal to 1,048,575.

Column: The **column** index. MUST be greater than or equal to 0 and less than or equal to 16,383.

Height: The number of rows. MUST be greater than or equal to 0. The sum of **Row** and **Height** MUST NOT be greater than 1,048,576.

Width: The number of columns. MUST be greater than or equal to 0. The sum of **Column** and **Width** MUST NOT be greater than 16,384.

2.2.4.5 Status

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

The status information returned by the protocol server defined as follows.

```
<xs:complexType name="Status" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Severity" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Name: The name of the status message. MUST be present. SHOULD be any subcode specified in section [2.2.9.1](#), but MAY be any other string.

Severity: The severity of the status message. MUST contain one of the values described in the following table.

Value	Meaning
Error	The status is of severity error.
Warning	The status is of severity warning.
Information	The status is of severity information.

Message: The message description.

2.2.4.6 TypeExporter

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

MUST NOT be used.

It is defined as follows.

```
<xs:complexType name="TypeExporter" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="CellError" type="tns:CellError"/>
  </xs:sequence>
</xs:complexType>
```

CellError: MUST NOT be used.

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
CellError	The CellError type is an enumeration of a set of cell error values that a worksheet cell can contain.
RangeReference	A string that is used by the protocol server to identify a range on a sheet (2) in a workbook , in some of the operations specified by this protocol.

Simple type	Description
SaveOptions	An enumeration of flags that can be specified for the SaveWorkbookCopy operation and is defined as follows.
SheetName	A string that is used by the protocol server to find a sheet (1) in a workbook in some of the operations specified in this protocol.
WorkbookType	An enumeration that is used in the GetWorkbook and SaveWorkbookCopy operations to specify the form of the requested workbook.

2.2.5.1 CellError

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

The **CellError** type is an enumeration of a set of **cell error values** that a **worksheet cell** can contain.

It is defined as follows.

```
<xs:simpleType name="CellError" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Div0"/>
    <xs:enumeration value="NA"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Null"/>
    <xs:enumeration value="Num"/>
    <xs:enumeration value="Ref"/>
    <xs:enumeration value="Value"/>
  </xs:restriction>
</xs:simpleType>
```

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

Value	Meaning
Div0	Worksheet cell contains a number that is divided by 0.
NA	Worksheet cell contains a value not available to a formula .
Name	Worksheet cell contains text not recognized by a formula.
Null	Worksheet cell contains an empty intersection of two areas.
Num	Worksheet cell contains invalid numeric values in a formula.
Ref	Worksheet cell contains a reference that is not valid.
Value	Worksheet cell contains a formula that includes worksheet cells that contain different data types.

2.2.5.2 RangeReference

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

A string that is used by the protocol server to identify a **range** on a **sheet (2)** in a **workbook**, in some of the operations specified by this protocol.

It has the following specification:

- MUST NOT be longer than 32768 characters.
- SHOULD use the A1 reference style as defined in [\[MSFT-ACRR\]](#), but MAY use any format supported by the server to identify the range.

This simple type is defined as follows.

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

2.2.5.3 SaveOptions

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

An enumeration of flags that can be specified for the **SaveWorkbookCopy** operation and is defined as follows.

```
<xs:simpleType name="SaveOptions" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:list>  
    <xs:simpleType>  
      <xs:restriction base="xs:string">  
        <xs:enumeration value="None"/>  
        <xs:enumeration value="AllowOverwrite"/>  
        <xs:enumeration value="LockAndCompareVersionsOnSave"/>  
      </xs:restriction>  
    </xs:simpleType>  
  </xs:list>  
</xs:simpleType>
```

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

Value	Meaning
None	MUST NOT be used.
AllowOverwrite	If this flag is not set, then if the workbookPath specified in the SaveWorkbookCopy operation corresponds to an existing file, the protocol server MUST NOT overwrite the existing file.
LockAndCompareVersionsOnSave	MUST NOT be used.

2.2.5.4 SheetName

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

A string that is used by the protocol server to find a **sheet (1)** in a **workbook** in some of the operations specified in this protocol.

It has the following specification:

- MUST NOT be longer than 31 characters.
- If empty, the protocol server MUST use the first sheet (1) in the workbook unless specified otherwise in the operation.

- If no sheet (1) in the workbook matches the requested **SheetName** simple type, the protocol server MUST return a SOAP fault response, which SHOULD contain the **InvalidSheetName** subcode but MAY contain any other subcode.

If the user does not have the **Open Item permission** for the workbook used by the **session (2)**, when **SheetName** is used by the **GetCell**, **GetCellA1**, **GetRange**, or **GetRangeA1** operations, the server MUST return a SOAP fault response, which MUST contain the **SpecifiedItemNotFound** subcode, if the matching sheet (1) is not a **visible** sheet (1).

This simple type is defined as follows.

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

2.2.5.5 WorkbookType

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

An enumeration that is used in the **GetWorkbook** and **SaveWorkbookCopy** operations to specify the form of the requested **workbook**.

This enumeration is defined as follows.

```
<xs:simpleType name="WorkbookType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FullWorkbook"/>
    <xs:enumeration value="FullSnapshot"/>
    <xs:enumeration value="PublishedItemsSnapshot"/>
  </xs:restriction>
</xs:simpleType>
```

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

Value	Meaning
FullWorkbook	The protocol server returns the workbook.
FullSnapshot	The protocol server returns a snapshot that contains the entire contents of the workbook.
PublishedItemsSnapshot	The protocol server returns a published items snapshot .

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.

2.2.9 Common Data Structures

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

If [\[SOAP1.2/1\]](#) or [\[SOAP1.2/2\]](#) is used for transport, the protocol server MUST use the **Value** element inside the **Subcode** element to pass the error code, as specified in [SOAP1.2/1], Section 5.4.1.3. Expected subcode values are specified in each operation. Common expected subcodes are specified in the [Common List of Subcodes](#).

2.2.9.1 Faults

The following table describes the error codes.

Value	Meaning
Blank	MUST NOT be used.
InternalServerError	An unexpected error occurred in the protocol server.
NotMemberOfRole	The user does not have permissions to complete the operation.
EncryptionTooLow	The protocol server is configured to accept only HTTPS , and the protocol client has tried to connect with a non HTTPS.
ConfirmDataRefresh	A connected data refresh confirmation is requested by the protocol client before connected data refresh on a workbook .
ExternalDataRefreshFailed	Connected data refresh failed.
ExternalDataRefreshFailedReal	Connected data refresh failed. <1>
ExternalDataRefreshFailedReadOrViewOnly	The query of one of the external sources failed (error message for users with read- or view-only permissions).
NoAccessToOdcReadOrViewOnly	The query of one of the external sources failed due to no access to the file in the format specified by [MS-ODCFF] (error message for users with read- or view-only permissions).
ErrorConnectingToDataSourceReadOrViewOnly	The query of one of the external sources failed due to failure to connect to the data source (error message for users with read- or view-only permissions).
InvalidUserNameOrPasswordReadOrViewOnly	The query of one of the external sources failed due to invalid credentials (error message for users with read- or view-only permissions).
InvalidUserNameOrPassword	The query of one of the external sources failed due to invalid credentials.
ExternalDataNotAllowedOnTL	The query of one of the external sources failed because external data connections are not allowed on the trusted location of the workbook session.

Value	Meaning
InvalidUriorLocalConnectionPath	The query of one of the external sources failed due to a local path for the file in the format specified by [MS-ODCFF].
DclNotTrusted	The query of one of the external sources failed because of the security settings on the protocol server.
OdcFileDoesNotExist	The query of one of the external sources failed because the specified file in the format specified by [MS-ODCFF] does not exist.
ErrorAccessingSsoFromEcs	The query of one of the external sources failed because the protocol server was unable to resolve the credentials of the protocol client.
SsoError	The query of one of the external sources failed because the protocol server was unable to resolve the credentials of the protocol client.
ErrorConnectingToDataSource	The query of one of the external sources failed due to the inability to connect to the specified data source.
UnattendedAccountNotConfigured	The query of one of the external sources failed because the protocol server was unable to authenticate against the data source.
IncorrectAuthUsed	The query of one of the external sources failed due to a credentials delegation error.
ExternalDataRefreshFailed_Details	Detail information regarding the connection failure.
PivotNoOverlapAllowed	A PivotTable grew to the point where it would overlap another PivotTable.
PivotOverlapList	A PivotTable grew to the point where it would overlap a list .
PivotCubeNotUnique	An OLAP query found more than one cube of the same name.
NoDataItemsOlap	An OLAP query found nothing to display.
OlapSchemaFail	An OLAP query was unable to fetch the schema of the cube.
InvalidPivot	A PivotTable is invalid.
OlapNoMoreDetail	A PivotTable has reached the lowest level of detail.
DrillActionInvalid	A PivotTable member cannot be expanded any further.
DriverNotOlap	The database does not support OLAP.
CubeMissing	The OLAP query was unable to locate the cube in the database.
SkipLine	Not all data received from the OLAP query could be displayed.
CDimMismatch	The OLAP data provider returned an unexpected number of dimensions .

Value	Meaning
DataSetAxisTooBig	The row or column area of the PivotTable is too large.
CubeIntegrityError	There was a problem with the structure of the OLAP cube.
PivotNotExpandedtoProtectCellContents	The PivotTable would have filled worksheet cells that already contain values, so the operation was canceled.
PivotTruncatedSheetEdge	The PivotTable was truncated by a worksheet edge.
DdlGrpNotSup	The PivotTable operation is not supported by the OLAP data provider.
PivotFormulasCollided	Some calculated PivotTable members were removed from the PivotTable because new PivotTable members of the same name appeared during refresh.
NoRetItems	The query of the PivotTable page field did not return any items.
OlapConnectVisualModeProvider	The PivotTable operation is not supported by the OLAP data provider.
PivotNoSourceData	The PivotTable was saved without the underlying data.
PivotFilteringError	The PivotTable could not be filtered.
PivotTableIncorrectDateFormat	An invalid date was used to filter the PivotTable.
PivotTableEndingLargerThanStarting	An invalid pair of starting and ending values was used to filter the PivotTable.
PivotTableIncorrectNumberFormat	A non-numeric value was used to filter the PivotTable, when a numeric value was required.
PivotTableOutOfRange	An out of range value was used to filter the PivotTable.
MergedCellOverlap	The PivotTable could not grow over part of a merged cell .
PivotSourceNotAvailable	The PivotTable could not be refreshed because its source is missing.
UnexpectedPivotError	The PivotTable has encountered an unexpected error.
EditModeCannotAcquireLockUnknownReason	An attempt to acquire or refresh a lock on a file failed.
EditModeCannotAcquireLock	An attempt to acquire or refresh a lock on a file failed because the file is already locked .
EditModeCannotAcquireLockUnknownUser	An attempt to acquire or refresh a lock on a file failed because the file is locked by an unknown user.
EditModeWorkbookCheckedOutRequired	An attempt to acquire or refresh a lock on a file failed because the file is not checked out .
EditModeWorkbookCheckedOutToLocalDrafts	An attempt to acquire or refresh a lock on a file failed because the file is checked out to a local drafts folder.

Value	Meaning
SiteTooBusy	MUST NOT be used.
SiteCollectionMemoryQuotaExceeded	MUST NOT be used.
SessionMemoryQuotaExceeded	MUST NOT be used.
SiteCollectionChartQuotaExceeded	MUST NOT be used.
RequestLimitsExceeded	Too many cells requested.
SavingUnsupportedInViewMode	An attempt was made to save a workbook from a view session .
SavingWorkbookNoWorkbookName	An attempt was made to save a workbook from a new workbook session .
FileSaveGeneric	The save operation failed.
SavingWorkbookInvalidPermissions	The user does not have permissions to save to the specified location.
SavingWorkbookMaximumSizeExceeded	An attempt to save a workbook failed because the workbook exceeds the maximum permitted size.
SavingWorkbookLockedOrCheckedOut	An attempt to save a workbook failed because the file is locked or checked out.
SavingWorkbookNotCheckedOut	An attempt to save a workbook failed because the file is not checked out.
SavingWorkbookAccessDenied	An attempt to save a workbook failed because the user does not have permission to overwrite an existing file.
SavingWorkbookInvalidFileType	MUST NOT be used.
SavingWorkbookFileNameExists	An attempt to save a workbook failed because the file already exists and the option to overwrite existing files was not specified.
SavingWorkbookNotTrustedLocation	An attempt to save a workbook failed because of the security settings on the protocol server.
SavingWorkbookInvalidLocation	An attempt to save a workbook failed because the specified file location is invalid.
SavingWorkbookInvalidFileName	An attempt to save a workbook failed because the specified file name is invalid.
SavingWorkbookTrustedLocationSettingWarning	An attempt to save a workbook failed because of the security settings on the protocol server.
SavingWorkbookAlmostTooLarge	MUST NOT be used.
SavingWorkbookTooLarge	MUST NOT be used.
NewWorkbookInvalidLocation	An attempt to create a new workbook failed because the specified location is invalid.
FileOpenSecuritySettings	Cannot open workbook because of security settings.

Value	Meaning
FileOpenNotFound	The specified workbook file does not exist.
FileOpenAccessDenied	The user does not have access to the workbook file.
FileCorrupt	The workbook file is in a format not recognized by the protocol server.
OpenWorkbookTooLarge	The protocol server failed to create a session (2) for a workbook because it is too large.
OpenWorkbookLocationNotTrusted	The protocol server failed to create a session (2) for a workbook because of the security settings on the protocol server.
OpenWorkbookAccessDenied	The protocol server failed to create a session (2) for a workbook because the user does not have sufficient permissions.
WorkbookNotSupported	The protocol server failed to create a session (2) for this workbook because it has unsupported content.
WorkbookNotSupportedViewRight	The protocol server failed to create a session (2) for this workbook because it has unsupported content and the user does not have Open Item permission .
UnsupportedFeatureViewWarning	MUST NOT be used.
UnsupportedFeatureSaveCopyWarning	MUST NOT be used.
ExternalWorkbookLinksDisabled	MUST NOT be used.
WorkbookLockedRead	An attempt to open a workbook failed because a file lock prevents it from being read.
EditModeCannotEditNonSharePointWorkbook	An attempt to open an edit session failed because the file is not in a document library .
EditModeAccessDenied	An attempt to open a workbook in an edit session failed because the user does not have sufficient permissions to edit the workbook file .
GenericFileOpenError	An unspecified error occurred when opening the workbook file.
IRMedWorkbook	An attempt to open a workbook failed because the workbook is in an Information Rights Management (IRM) document library.
ParameterInvalid	An attempt to open a workbook failed because the workbook contains one or more invalid workbook parameters .
ParameterLimitExceeded	An attempt to open a workbook failed because the workbook contains too many workbook parameters.
WorkbookInternalLimit	The workbook cannot be loaded by the protocol server because of a file-size limitation.
OldFormatConversionFailed	MUST NOT be used.

Value	Meaning
EditModeWorkbookExternalData	An attempt to open a workbook as an edit session failed because it contains connected data or user-defined functions (UDFs) , and the workbook file has not been checked out.
FormulaEditingNotEnabled	The operation failed because it tried to set a formula into a workbook parameter.
DimensionAndArrayMismatch	An attempt to set a range failed because the dimensions of the specified array of cells do not match the specified range.
SetRangeFailure	The operation failed to set a value into a cell.
ObjectTypeNotSupported	The operation failed because it tried to update a cell with an object of an unsupported type.
ParameterDoesNotExist	The specified workbook parameter does not exist.
FormulaParseFailed	The specified formula could not be parsed.
SetParameterFailure	An unspecified error occurred when attempting to set a workbook parameter.
CannotEditPivotTable	The operation failed because it tried to set a cell in a PivotTable.
InsDelCellsNonblankCells	The operation failed because it would require shifting non-blank cells off the sheet (1) .
InsDelCellsArrayFmla	The operation failed because it tried to change part of an array formula .
InsDelCellsPivotTable	The operation failed because it tried to change part of a PivotTable.
InsDelCellsTable	The operation failed because it tried to change part of a table .
InsDelCellsMergedCells	The operation failed because it tried to change part of a merged cell.
InsDelCellsDatatable	The operation failed because it tried to change part of a data table .
InsDelCellsRowsForWholeColumn	The operation failed because it tried to insert or delete a whole row, but the specified range is a whole column.
InsDelCellsColumnsForWholeRow	The operation failed because it tried to insert or delete a whole column, but the specified range is a whole row.
InvalidSheetName	The operation failed because the specified sheet (1) name does not exist.
RangeParseError	The operation failed because the specified range is invalid.
SpecifiedRangeNotFound	The operation failed because the specified range was not found.
RangeDefinitionError	The operation failed because the specified range is not a valid sheet (2) location.
DiscontiguousRangeNotSupported	The operation failed because a non-contiguous range was specified.

Value	Meaning
RangeRequestAreaExceeded	The requested range was too large.
SheetRangeMismatch	The sheet (2) provided as the sheet (2) argument is not the same as the sheet (2) specified in the range argument.
ImageGenFailed	The protocol server failed to create an image.
ApiInvalidArgument	The operation failed because an argument that passed was invalid.
ApiInvalidCoordinate	MUST NOT be used.
ConfirmationUnsupported	MUST NOT be used.
ApiInvalidDimensionReturned	MUST NOT be used.
ApiMissingSiteContext	MUST NOT be used.
Ewr_IllegalPageSize	MUST NOT be used.
Ewr_WorkbookNotSpecified	MUST NOT be used.
Ewr_LoadProgress	MUST NOT be used.
Ewr_FileCorrupt	MUST NOT be used.
Ewr_WorkbookNotSupported	MUST NOT be used.
Ewr_GenericFileOpenError	MUST NOT be used.
Ewr_InvalidParameterGeneric	MUST NOT be used.
Ewr_InvalidParameterMandatoryMissing	MUST NOT be used.
Ewr_InvalidParameterUnexpectedFormat	MUST NOT be used.
Ewr_InvalidParameterOutOfRange	MUST NOT be used.
Ewr_InvalidParameterOperationNotAllowed	MUST NOT be used.
Ewr_ParameterInvalid	MUST NOT be used.
Ewr_ParameterLimitExceeded	MUST NOT be used.
Ewr_ParameterDoesNotExist	MUST NOT be used.
Ewr_SetParameterFailure	MUST NOT be used.

Value	Meaning
Ewr_SetParameterFailureOnOpen	MUST NOT be used.
Ewr_ParameterReadFromDocPropsError	MUST NOT be used.
Ewr_FormulaEditingNotEnabled	MUST NOT be used.
Ewr_InvalidVisibleItem	MUST NOT be used.
Ewr_EditSaveAsOverCurrentWorkbook	MUST NOT be used.
Ewr_RequestTimedOut	MUST NOT be used.
Ewr_ServerCommunicationError	MUST NOT be used.
Ewr_ServerAuthenticationError	MUST NOT be used.
Mewa_PageStateInitFailed	MUST NOT be used.
Mewa_CriticalParameterMissing	MUST NOT be used.
Mewa_GridViewRenderingFailed	MUST NOT be used.
Mewa_FeatureNotActive	MUST NOT be used.
OperationNotSupportedInViewOnly	The operation failed because it is not supported in a view session.
OperationOnInvalidState	An attempt was made to open a session (2) that already has a workbook assigned.
InvalidOrTimedOutSession	The operation failed because the session (2) is invalid or timed out.
ScratchPadChangesLost	A view session that is not in view-only mode timed out while the user was attempting to convert it to an edit session.
MaxSessionsPerUserExceeded	The user has exceeded the maximum allowed session (2) count by the protocol server.
OperationCanceled	The user canceled a request that was being processed by a session (2).
MaxRequestDurationExceeded	The request failed because it took too long to process.
NoOccurrencesFound	The protocol server could not match the sought string to any worksheet cell relevant to the search.
EcsInvalidNamedObject	An invalid named object was requested.
NoPublishedItems	The workbook does not have any published items .
RetryError	The operation could not be finished at this time, but the request may succeed if it is sent again by the protocol client.

Value	Meaning
StopOpenOnRefreshFailure	An attempt to open a workbook failed because connected data refresh failed.
ProductExpired	The protocol server returns this error when the application license key is invalid.
SortActionInvalid	Sorting an AutoFilter failed because it contains merged cells that are not all the same size.
SortActionNotSupportedWithPivot	Sorting an AutoFilter failed because it contains a PivotTable.
CircularReferencesFound	MUST NOT be used.
SessionNotFound	MUST NOT be used.
NewWorkbookSessionTimeout	A new workbook session has timed out.
EditXLSFile	MUST NOT be used.
UnsupportedFeatureEditWarning	Edit session is not supported for this workbook.
EditModeSessionExists	The view session could not be saved because the workbook is open in an edit session.
NewerWorkbookExists	The view session could not be saved because the workbook has been modified since the session (2) was opened.
ListEntryNotFound_1	MUST NOT be used.
ListEntryNotFound_2	MUST NOT be used.
ListEntryAlreadyExists_1	MUST NOT be used.
ListEntryAlreadyExists_2	MUST NOT be used.
InvalidArgumentValue	MUST NOT be used.
MissingArgumentName	MUST NOT be used.
InvalidArgumentName	MUST NOT be used.
ArgumentBadInRemove	MUST NOT be used.
ManagementUIListEntryAlreadyExists	MUST NOT be used.
ManagementUIListEntryNotFound	MUST NOT be used.
ManagementUIInternalError	MUST NOT be used.
UnsupportedUICulture	The UI culture is not supported by the protocol server.

Value	Meaning
UnsupportedDataCulture	The data culture is not supported by the protocol server.
SpecifiedItemNotFound	An attempt to set a range of cells in a workbook failed because the user does not have Open Item permission.
EcsSnapshotAccessDenied	The protocol server cannot return the workbook or a snapshot , because the user does not have Open Item permission.
SaveFailed	A save operation failed.
WrongEcs	An attempt was made to access an edit session that is owned by another instance of a protocol server in the farm .
PasteDataLimitHit	The string exceeds the maximum permitted length.
UndoRedoFail	MUST NOT be used.
CollabLocaleMismatch	The user joined an edit session whose data culture does not match the user's data culture.
LocaleRestored	The data culture of an edit session has been restored to the user's data culture.
InvalidEditSession	The edit session is invalid.
InvalidEditSessionId	MUST NOT be used.
UnableToLoadEditSession	Unable to load an edit session.
SavingWorkbookFailed	Failed to save workbook.
RefreshLockFailed	An attempt to refresh a lock on a workbook file failed.
InvalidChartName	An invalid chart name was specified.
RetryOutOfSync	The PivotTable operation failed.
OperationFailed	A workbook operation failed.
OpenInExcelActiveXError	MUST NOT be used.
Count	MUST NOT be used.

2.2.9.2 Common List of Subcodes

In the event of an application error during an operation the protocol server returns a SOAP fault response, which SHOULD [<2>](#) contain one of the following subcodes:

Subcode
InternalError
OperationFailed
NotMemberOfRole
EncryptionTooLow
ConfirmDataRefresh
OperationOnInvalidState
InvalidOrTimedOutSession
MaxSessionsPerUserExceeded
OperationCanceled
MaxRequestDurationExceeded
RetryError
ProductExpired
NewWorkbookSessionTimeout
SpecifiedItemNotFound
CollabLocaleMismatch
InvalidEditSession
UnableToLoadEditSession
SpecifiedRangeNotFound
EditModeAccessDenied
EditModeCannotAcquireLockUnknownReason
RequestLimitsExceeded
SavingWorkbookFailed
RefreshLockFailed
ApiInvalidArgument

3 Protocol Details

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

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

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

This protocol allows protocol servers to perform implementation-specific localization of text in various messages. Except where specified, the localization of this text is an implementation-specific behavior of the protocol server and not significant for interoperability.

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

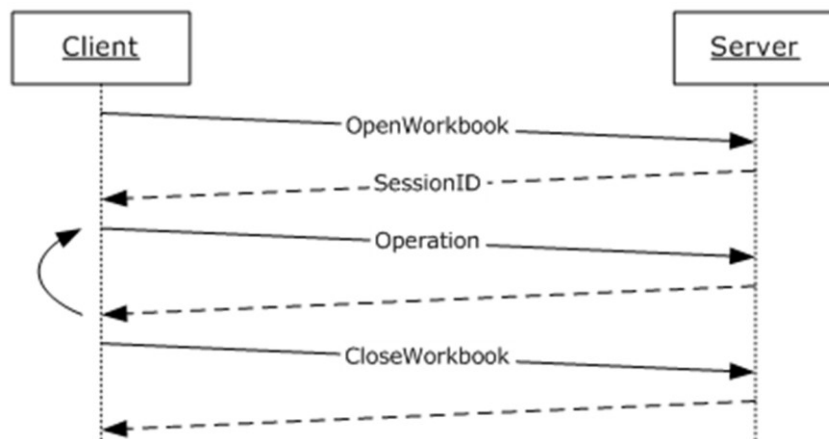


Figure 2: High-level sequence diagram

Session Behavior

View Sessions

A **view session** is created on a successful **OpenWorkbook** or **OpenWorkbookEx** request. A view session stops responding to operations after a successful call to the **CloseWorkbook** operation or after the view session has timed out.

New Workbook Sessions

A **new workbook session** is created on a successful **NewWorkbook** request. A new workbook session stops responding to operations after a successful call to the **CloseWorkbook** operation or after the new workbook session has timed out.

Edit Sessions

An **edit session** is created on a successful **OpenWorkbookForEditing** request, unless one already exists for the same **workbookPath**, as specified in **OpenWorkbookForEditing** (section 3.1.4.21). A new participant is added to the edit session on a successful **OpenWorkbookForEditing** request.

A participant that makes a call to **CloseWorkbook** is removed from the edit session if the operation completes successfully. An edit session stops responding to operations when all of its participants are removed.

Request Behavior

A request is created when a **Web service** operation starts running on the server, and ends when the Web service operation finishes, when it is successfully cancelled by a **CancelRequest** call, or after the request has timed out.

3.1 Server Details

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.

Session: An entity that gives context to operations done against a workbook. A **session (2)** is associated with a workbook state, which can be modified by state changing requests.

Request: An entity that represents a running request on the protocol server.

3.1.2 Timers

3.1.2.1 Session Timeout

The protocol server MAY<3> time **sessions (2)** out. When a session (2) times out, the protocol server MUST NOT serve any requests to the timed out session (2).

3.1.2.2 Request Timeout

The protocol server MAY<4> time requests out. When a request times out, the protocol server SHOULD<5> return a **SOAP fault** response, with the subcode **MaxRequestDurationExceeded**.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
Calculate	This operation is used to calculate a specific range in the workbook .
CalculateA1	This operation is used to calculate a specific range in the workbook.
CalculateWorkbook	This operation is used to calculate the workbook.

Operation	Description
CancelRequest	This operation is used to cancel an operation that is currently running on the protocol server.
CloseWorkbook	This operation is used to close an existing session (2) or leave an edit session on the protocol server.
EnsureWorkbookModel	Loads workbook data model and returns information about the model.
GetApiVersion	This operation is used to get the version of the protocol server.
GetCell	This operation is used to retrieve the contents of a specific cell in the workbook.
GetCellA1	This operation is used to retrieve the contents of a specific cell in the workbook.
GetChartImageUrl	This operation returns the URL of a chart in a workbook.
GetPublishedItemNames	This operation returns a list of the published items in a workbook.
GetRange	This operation is used to retrieve the contents of a specific range in the workbook.
GetRangeA1	This operation is used to retrieve the contents of a named range in the workbook.
GetSessionInformation	This operation retrieves information about a session (2) on the protocol server.
GetSheetNames	The operation returns information about the sheets in a workbook.
GetWorkbook	This operation is used to retrieve a workbook file.
GetWorkbookConnections	Returns a list of data connection names in the workbook.
NewWorkbook	This operation is used to create a new workbook session .
OpenWorkbook	This operation is used to create a new view session , giving access to a workbook on the protocol server.
OpenWorkbookEx	This operation is used to create a new view session, giving access to a workbook on the protocol server.
OpenWorkbookForEditing	This operation is used to create a new edit session, giving access to a workbook on the protocol server.
Refresh	This operation is used to refresh connected data on a workbook.
RefreshDetached	MUST NOT be used.
RefreshEx	This operation is used to refresh connected data on a workbook.
SaveWorkbook	This operation is used to save a workbook associated with an open session (2) to the same location from which it was opened.
SaveWorkbookCopy	This operation is used to save a workbook associated with an open session (2) to a specified location, and in a specified format.
SaveWorkbookCopyDetached	MUST NOT be used.
SetCalculationOptions	This operation is used to set the workbook calculation mode .
SetCell	This operation is used to set a value to a specific worksheet cell in the

Operation	Description
	workbook.
SetCellA1	This operation is used to set a value to a specific worksheet cell in the workbook.
SetParameters	This operation to sets values of workbook parameters in a workbook.
SetRange	This operation is used to set values to a range of worksheet cells on a workbook.
SetRangeA1	This operation is used to set values to a range of worksheet cells on a workbook.

3.1.4.1 Calculate

This operation is used to calculate a specific **range** in the **workbook**.

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

```
<wsdl:operation name="Calculate" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CalculateSoapIn"/>
  <wsdl:output message="tns:CalculateSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **CalculateSoapIn** request message, and the protocol server responds with a **CalculateSoapOut** response message as follows:

1. If the protocol determines that the workbook does not need to be **calculated**, it MUST succeed the operation and it MUST ignore the **sheetName** and **rangeCoordinates** parameters.
2. **sheetName** MUST NOT be empty if **rangeCoordinates** is not empty, as defined in [RangeCoordinates](#). If **sheetName** is empty, the whole workbook MUST be calculated.
3. If **rangeCoordinates** is empty, as defined in section 2.2.4.4, and the **sheetName** is not empty, the whole **sheet (2)**, which was specified in **sheetName**, MUST be calculated.
4. If **rangeCoordinates** is not empty, the specified range MUST be calculated.
5. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError

Subcode
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.1.1 Messages

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

Message	Description
CalculateSoapIn	The request WSDL message for the Calculate WSDL operation .
CalculateSoapOut	The response WSDL message for the Calculate WSDL operation .

3.1.4.1.1.1 CalculateSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/Calculate
```

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

3.1.4.1.1.2 CalculateSoapOut

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

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

3.1.4.1.2 Elements

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

Element	Description
Calculate	The input data for the Calculate WSDL operation .
CalculateResponse	The result data for the Calculate WSDL operation .

3.1.4.1.2.1 Calculate

The **Calculate** element specifies the input data for the **Calculate WSDL operation**.

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

```

<xs:sequence>
  <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
  <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
  <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
</xs:sequence>
</xs:complexType>
</xs:element>

```

sessionId: A **session identifier**. MUST be present in the request message. MUST NOT be longer than 1023 characters. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** to calculate. MUST be present. MUST NOT be longer than 31 characters.

rangeCoordinates: The range to calculate on the specified sheet (2). MUST be present.

3.1.4.1.2.2 CalculateResponse

The **CalculateResponse** element specifies the result data for the **Calculate WSDL operation**.

```

<xs:element name="CalculateResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

status: An array of status information as specified in **ArrayOfStatus** (section [2.2.4.2](#)).

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 CalculateA1

This operation is used to **calculate** a specific range in the **workbook**.

The following is the **WSDL** port type specification of the **CalculateA1 WSDL operation**.

```

<wsdl:operation name="CalculateA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CalculateA1SoapIn"/>
  <wsdl:output message="tns:CalculateA1SoapOut"/>
</wsdl:operation>

```

The protocol client sends a **CalculateA1SoapIn** request message, and the protocol server responds with a **CalculateA1SoapOut** response message, as follows:

1. If the protocol determines that the workbook does not need to be calculated, it MUST succeed the operation and it MUST ignore the **sheetName** and **rangeName** parameters.
2. **sheetName** MUST be present and MUST NOT be empty, if **rangeName** is not empty. If **sheetName** is not present or is empty, the whole workbook MUST be calculated.
3. **rangeName** MUST be present. If **rangeName** is empty, and **sheetName** is present and not empty, the whole **sheet (2)**, which was specified in **sheetName**, MUST be calculated.
4. If **rangeName** is not empty, the specified **range** MUST be calculated.
5. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.2.1 Messages

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

Message	Description
CalculateA1SoapIn	The request WSDL message for the CalculateA1 WSDL operation .
CalculateA1SoapOut	The response WSDL message for the CalculateA1 WSDL operation .

3.1.4.2.1.1 CalculateA1SoapIn

The request **WSDL message** for the **CalculateA1 WSDL operation**.

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/CalculateA1
```

The **SOAP body** contains the **CalculateA1** element.

3.1.4.2.1.2 CalculateA1SoapOut

The response **WSDL message** for the **CalculateA1 WSDL operation**.

The **SOAP body** contains the **CalculateA1Response** element.

3.1.4.2.2 Elements

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

Element	Description
CalculateA1	The input data for the CalculateA1 WSDL operation .
CalculateA1Response	The result data for the CalculateA1 WSDL operation .

3.1.4.2.2.1 CalculateA1

The **CalculateA1** element specifies the input data for the **CalculateA1 WSDL operation**.

```
<xs:element name="CalculateA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session (2)** identifier. MUST be present in the request message. MUST NOT be longer than 1023 characters. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** to calculate. MUST be present and MUST NOT be empty, if **rangeName** is not empty. MUST NOT be longer than 31 characters.

rangeName: The name of the range to calculate. MUST be present. MUST NOT be longer than 255 characters. See section [2.2.5.2](#).

3.1.4.2.2.2 CalculateA1Response

The **CalculateA1Response** element specifies the result data for the **CalculateA1 WSDL operation**.

```
<xs:element name="CalculateA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

status: An array of status information as specified in section [2.2.4.2](#).

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 CalculateWorkbook

This operation is used to **calculate** the **workbook**.

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

```

<wsdl:operation name="CalculateWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CalculateWorkbookSoapIn"/>
  <wsdl:output message="tns:CalculateWorkbookSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **CalculateWorkbookSoapIn** request message and the protocol server responds with a **CalculateWorkbookSoapOut** response message as follows:

If there was an application error on the protocol server during the operation, the protocol server **MUST** return a **SOAP fault** response, which **SHOULD** contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but **MAY** instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject

Subcode
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.3.1 Messages

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

Message	Description
CalculateWorkbookSoapIn	The request WSDL message for the CalculateWorkbook WSDL operation .
CalculateWorkbookSoapOut	The response WSDL message for the CalculateWorkbook WSDL operation .

3.1.4.3.1.1 CalculateWorkbookSoapIn

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

The **SOAP action** value is:

`http://schemas.microsoft.com/office/excel/server/webservices/CalculateWorkbook`

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

3.1.4.3.1.2 CalculateWorkbookSoapOut

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

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

3.1.4.3.2 Elements

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

Element	Description
CalculateWorkbook	The input data for the CalculateWorkbook WSDL operation .
CalculateWorkbookResponse	The result data for the CalculateWorkbook WSDL operation .

3.1.4.3.2.1 CalculateWorkbook

The **CalculateWorkbook** element specifies the input data for the **CalculateWorkbook WSDL operation**.

```
<xs:element name="CalculateWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="calculateType" type="tns:CalculateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session (2)** identifier. MUST be present in the request message. MUST NOT be longer than 1023 characters. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

calculateType: The type of calculation to perform, as defined in section [3.1.4.3.4.1](#).

3.1.4.3.2.2 CalculateWorkbookResponse

The **CalculateWorkbookResponse** element specifies the result data for the **CalculateWorkbook WSDL operation**.

```
<xs:element name="CalculateWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

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

Simple type	Description
CalculateType	The type of calculation to be executed.

3.1.4.3.4.1 CalculateType

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

The type of calculation to be executed.

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

```

<xs:restriction base="xs:string">
  <xs:enumeration value="Recalculate"/>
  <xs:enumeration value="CalculateFull"/>
</xs:restriction>
</xs:simpleType>

```

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

Value	Meaning
Recalculate	The protocol server MUST calculate all formulas in the workbook that need to be calculated.
CalculateFull	The protocol server MUST calculate all formulas.

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 CancelRequest

This operation is used to cancel an operation that is currently running on the protocol server.

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

```

<wsdl:operation name="CancelRequest" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CancelRequestSoapIn"/>
  <wsdl:output message="tns:CancelRequestSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **CancelRequestSoapIn** request message, and the protocol server responds with a **CancelRequestSoapOut** response message, as follows:

- The protocol server SHOULD try to cancel all requests currently running on the protocol server against the **session (2)** specified in **sessionId**.
- The following MUST NOT cause an application error in the **CancelRequest** operation:
 - An operation is not currently running on the session (2).
 - A currently running operation on the session (2) cannot be cancelled.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.4.1 Messages

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

Message	Description
CancelRequestSoapIn	The request WSDL message for the CancelRequest WSDL operation .
CancelRequestSoapOut	The response WSDL message for the CancelRequest WSDL operation .

3.1.4.4.1.1 CancelRequestSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/CancelRequest
```

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

3.1.4.4.1.2 CancelRequestSoapOut

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

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

3.1.4.4.2 Elements

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

Element	Description
CancelRequest	The input data for the CancelRequest WSDL operation .
CancelRequestResponse	The result data for the CancelRequest WSDL operation .

3.1.4.4.2.1 CancelRequest

The **CancelRequest** element specifies the input data for the **CancelRequest WSDL operation**.

```
<xs:element name="CancelRequest" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session identifier** MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, or but MAY instead be obtained in any other way.

3.1.4.4.2 CancelRequestResponse

The **CancelRequestResponse** element specifies the result data for the **CancelRequest WSDL operation**.

```
<xs:element name="CancelRequestResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in **ArrayOfStatus** (section [2.2.4.2](#)).

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 CloseWorkbook

This operation is used to close an existing **session (2)** or leave an **edit session** on the protocol server.

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

```
<wsdl:operation name="CloseWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:CloseWorkbookSoapIn"/>
  <wsdl:output message="tns:CloseWorkbookSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **CloseWorkbookSoapIn** request message and the protocol server responds with a **CloseWorkbookSoapOut** response message as follows:

If there was an application error on the protocol server during the operation, the protocol server **MUST** return a SOAP fault response, which **SHOULD** be one of the subcodes specified in section [2.2.9.2](#), but **MAY** instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.5.1 Messages

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

Message	Description
CloseWorkbookSoapIn	The request WSDL message for the CloseWorkbook WSDL operation .
CloseWorkbookSoapOut	The response WSDL message for the CloseWorkbook WSDL operation .

3.1.4.5.1.1 CloseWorkbookSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/CloseWorkbook
```

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

3.1.4.5.1.2 CloseWorkbookSoapOut

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

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

3.1.4.5.2 Elements

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

Element	Description
CloseWorkbook	The input data for the CloseWorkbook WSDL operation .
CloseWorkbookResponse	The result data for the CloseWorkbook WSDL operation .

3.1.4.5.2.1 CloseWorkbook

The **CloseWorkbook** element specifies the input data for the **CloseWorkbook WSDL operation**.

```
<xs:element name="CloseWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session identifier**. MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, or but MAY instead be obtained in any other way.

3.1.4.5.2 CloseWorkbookResponse

The **CloseWorkbookResponse** element specifies the result data for the **CloseWorkbook WSDL operation**.

```
<xs:element name="CloseWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: MUST NOT be present.

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 EnsureWorkbookModel

Loads workbook data model and returns information about the model.

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

```
<wsdl:operation name="EnsureWorkbookModel" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:EnsureWorkbookModelSoapIn"/>
  <wsdl:output message="tns:EnsureWorkbookModelSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **EnsureWorkbookModelSoapIn** request message, and the protocol server responds with a **EnsureWorkbookModelSoapOut** response message as follows:

- The protocol server loads workbook data model and returns information about the loaded model in the **WorkbookModelInfo** element.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.6.1 Messages

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

Message	Description
EnsureWorkbookModelSoapIn	The request WSDL message for the EnsureWorkbookModel WSDL operation .
EnsureWorkbookModelSoapOut	The response WSDL message for the EnsureWorkbookModel WSDL operation .

3.1.4.6.1.1 EnsureWorkbookModelSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/EnsureWorkbookModel
```

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

3.1.4.6.1.2 EnsureWorkbookModelSoapOut

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

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

3.1.4.6.2 Elements

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

Element	Description
EnsureWorkbookModel	The input data for the EnsureWorkbookModel WSDL operation .
EnsureWorkbookModelResponse	The result data for the EnsureWorkbookModel WSDL operation .

3.1.4.6.2.1 EnsureWorkbookModel

The **EnsureWorkbookModel** element specifies the input data for the **EnsureWorkbookModel WSDL operation**.

```
<xs:element name="EnsureWorkbookModel" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```


sessionId: A **session identifier**. MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

3.1.4.6.2.2 EnsureWorkbookModelResponse

The **EnsureWorkbookModelResponse** element specifies the result data for the **EnsureWorkbookModel WSDL operation**.

```
<xs:element name="EnsureWorkbookModelResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="EnsureWorkbookModelResult"
type="tns:WorkbookModelInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

EnsureWorkbookModelResult: **WorkbookModelInfo** element as specified in section [3.1.4.6.3.1](#)

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.6.3 Complex Types

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

Complex type	Description
WorkbookModelInfo	Specifies server, database, version of the data model, and session timeout (section 3.1.2.1).

3.1.4.6.3.1 WorkbookModelInfo

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

Specifies **server**, **database**, **version** of the data model, and session timeout (section [3.1.2.1](#)).

```
<xs:complexType name="WorkbookModelInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Server" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Database" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="SessionInteractivityTimeout"
type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Version" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

Server: The server where the data model was loaded.

Database: The database on the server where the data model was loaded.

SessionInteractivityTimeout: Session timeout (section 3.1.2.1).

Version: Version of the data model.

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 GetApiVersion

This operation is used to get the version of the protocol server.

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

```
<wsdl:operation name="GetApiVersion" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetApiVersionSoapIn"/>
  <wsdl:output message="tns:GetApiVersionSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetApiVersionSoapIn** request message, and the protocol server responds with a **GetApiVersionSoapOut** response message as follows:

- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.7.1 Messages

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

Message	Description
GetApiVersionSoapIn	The request WSDL message for the GetApiVersion WSDL operation .
GetApiVersionSoapOut	The response WSDL message for the GetApiVersion WSDL operation .

3.1.4.7.1.1 GetApiVersionSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetApiVersion
```

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

3.1.4.7.1.2 GetApiVersionSoapOut

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

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

3.1.4.7.2 Elements

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

Element	Description
GetApiVersion	The input data for the GetApiVersion WSDL operation .
GetApiVersionResponse	The result data for the GetApiVersion WSDL operation .

3.1.4.7.2.1 GetApiVersion

The **GetApiVersion** element specifies the input data for the **GetApiVersion WSDL operation**.

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

3.1.4.7.2.2 GetApiVersionResponse

The **GetApiVersionResponse** element specifies the result data for the **GetApiVersion WSDL operation**.

```
<xs:element name="GetApiVersionResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetApiVersionResult" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetApiVersionResult: The version of the server. MUST be present.

status: MUST NOT be present.

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 GetCell

This operation is used to retrieve the contents of a specific **cell** in the **workbook**.

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

```
<wsdl:operation name="GetCell" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetCellSoapIn"/>
  <wsdl:output message="tns:GetCellSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetCellSoapIn** request message, and the protocol server responds with a **GetCellSoapOut** response message, as follows:

1. The **row**, **column** on the **worksheet** specified by **sheetName**, specify the location of the **sheet (2)** cells to retrieve.
2. If **formatted** is **false**, the worksheet cell returned as the **GetCellResult** element MUST be either not present or one of the following types: **boolean**, **double**, **string**, or **CellError** (section [2.2.5.1](#)).
3. If **formatted** is true, the worksheet cell returned as the **GetCellResult** element MUST be type **string**, and MUST be formatted using a **number format** according to how the worksheet cell is number formatted in the workbook.
4. If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.8.1 Messages

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

Message	Description
GetCellSoapIn	The request WSDL message for the GetCell WSDL operation .
GetCellSoapOut	The response WSDL message for the GetCell WSDL operation .

3.1.4.8.1.1 GetCellSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetCell
```

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

3.1.4.8.1.2 GetCellSoapOut

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

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

3.1.4.8.2 Elements

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

Element	Description
GetCell	The input data for the GetCell WSDL operation .
GetCellResponse	The result data for the GetCell WSDL operation .

3.1.4.8.2.1 GetCell

The **GetCell** element specifies the input data for the **GetCell WSDL operation**.

```
<xs:element name="GetCell" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **worksheet** that contains the cell. See section [2.2.5.4](#). MUST be present. MUST NOT be empty.

row: The **row** index of the **cell**. MUST be greater than or equal to 0 and less than or equal to 1,048,575.

column: The **column** index of the cell. MUST be greater than or equal to 0 and less than or equal to 16,383

formatted: Specifies whether **GetCellResult** will be number formatted.

3.1.4.8.2.2 GetCellResponse

The **GetCellResponse** element specifies the result data for the **GetCell WSDL operation**.

```
<xs:element name="GetCellResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCellResult"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetCellResult: The returned cell value.

status: An array of status information as specified in section [2.2.4.2](#).

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 GetCellA1

This operation is used to retrieve the contents of a specific **cell** in the **workbook**.

The following is the **WSDL** port type specification of the **GetCellA1 WSDL operation**.

```

<wsdl:operation name="GetCellA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetCellA1SoapIn"/>
  <wsdl:output message="tns:GetCellA1SoapOut"/>
</wsdl:operation>

```

The protocol client sends a **GetCellA1SoapIn** request message, and the protocol server responds with a **GetCellA1SoapOut** response message, as follows:

1. The **rangeName** specifies the location of the **sheet (2)** cell to retrieve. The sheet (2) that contains the sheet (2) cell to retrieve is determined in the following manner.
 - If **rangeName** specifies a sheet (2), that is the sheet (2) which MUST be used.
 - Otherwise, the sheet (2) specified in **sheetName** MUST be used, as specified in section [2.2.5.4](#).
 - If neither **sheetName** nor **rangeName** specify a sheet (2), the protocol server MAY [<6>](#) use any sheet (1) in the workbook.
2. If **formatted** is false, the sheet (2) cell returned as the **GetCellA1Result** element MUST be either not present or one of the following types: **boolean**, **double**, **string**, or **CellError** (section [2.2.5.1](#)).
3. If **formatted** is true, the sheet (2) cell returned as the **GetCellA1Result** element MUST be type **string**, and MUST be number formatted according to how the sheet (2) cell is number formatted in the workbook.
4. If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.9.1 Messages

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

Message	Description
GetCellA1SoapIn	The request WSDL message for the GetCellA1 WSDL operation .
GetCellA1SoapOut	The response WSDL message for the GetCellA1 WSDL operation .

3.1.4.9.1.1 GetCellA1SoapIn

The request **WSDL message** for the **GetCellA1 WSDL operation**.

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetCellA1
```

The **SOAP body** contains the **GetCellA1** element.

3.1.4.9.1.2 GetCellA1SoapOut

The response **WSDL message** for the **GetCellA1 WSDL operation**.

The **SOAP body** contains the **GetCellA1Response** element.

3.1.4.9.2 Elements

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

Element	Description
GetCellA1	The input data for the GetCellA1 WSDL operation .
GetCellA1Response	The result data for the GetCellA1 WSDL operation .

3.1.4.9.2.1 GetCellA1

The **GetCellA1** element specifies the input data for the **GetCellA1 WSDL operation**.

```
<xs:element name="GetCellA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** that contains the **cell**. See section [2.2.5.4](#) for specification. MUST be present.

rangeName: The name of the **range** as specified in section [2.2.5.2](#). MUST be present. MUST NOT be longer than 255 characters.

formatted: Specifies whether **GetCellA1Result** will be number formatted.

3.1.4.9.2.2 GetCellA1Response

The **GetCellA1Response** element specifies the result data for the **GetCellA1 WSDL operation**.

```
<xs:element name="GetCellA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCellA1Result"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetCellA1Result: The returned cell value.

status: An array of status information as specified in section [2.2.4.2](#).

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 GetChartImageUrl

This operation returns the **URL** of a chart in a **workbook**.

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

```
<wsdl:operation name="GetChartImageUrl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetChartImageUrlSoapIn"/>
  <wsdl:output message="tns:GetChartImageUrlSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetChartUrlSoapIn** request message, and the protocol server responds with a **GetChartUrlSoapOut** response message as follows:

- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.10.1 Messages

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

Message	Description
GetChartImageUrlSoapIn	The request WSDL message for the GetChartImageUrl WSDL operation .
GetChartImageUrlSoapOut	The response WSDL message for the GetChartImageUrl WSDL operation .

3.1.4.10.1.1 GetChartImageUrlSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl
```

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

3.1.4.10.1.2 GetChartImageUrlSoapOut

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

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

3.1.4.10.2 Elements

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

Element	Description
GetChartImageUrl	The input data for the GetChartImageUrl WSDL operation .
GetChartImageUrlResponse	The result data for the GetChartImageUrl WSDL operation .

3.1.4.10.2.1 GetChartImageUrl

The **GetChartImageUrl** element specifies the input data for the **GetChartImageUrl WSDL operation**.

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

```

    <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="size" type="tns:Size"/>
    <xs:element minOccurs="0" maxOccurs="1" name="chartName" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

sessionId: A **session identifier** . MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

size: The size of the chart image as defined in section [3.1.4.10.3.1](#). If not present, the original chart size will be used.

chartName: The string identifier for a chart. MUST be present. MUST be non-empty.

3.1.4.10.2.2 GetChartImageUrlResponse

The **GetChartImageUrlResponse** element specifies the result data for the **GetChartImageUrl WSDL operation**.

```

<xs:element name="GetChartImageUrlResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetChartImageUrlResult"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

GetChartImageUrlResult: The **URL** of the requested chart.

status: An array of **Status** elements as specified in section [2.2.4.2](#).

3.1.4.10.3 Complex Types

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

Complex type	Description
Size	Specifies the size of a chart image.

3.1.4.10.3.1 Size

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

Specifies the size of a chart image defined as follows.

```

<xs:complexType name="Size" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

```

Width: Specifies the width of the image to be returned. MUST be an integer greater than or equal to 4 and less than or equal to 4096, or -1. If -1, the original width of the chart will be used. Must be -1 if **Height** is -1.

Height: Specifies the height of the image to be returned. MUST be an integer greater than or equal to 4 and less than or equal to 4096, or -1. If -1, the original height of the chart will be used. MUST be -1 if **Width** is -1.

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 GetPublishedItemNames

This operation returns a list of the **published items** in a **workbook**.

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

```
<wsdl:operation name="GetPublishedItemNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetPublishedItemNamesSoapIn"/>
  <wsdl:output message="tns:GetPublishedItemNamesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetPublishedItemNamesSoapIn** request message, and the protocol server responds with a **GetPublishedItemNamesSoapOut** response message as follows:

- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.11.1 Messages

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

Message	Description
GetPublishedItemNamesSoapIn	The request WSDL message for the GetPublishedItemNames WSDL operation .
GetPublishedItemNamesSoapOut	The response WSDL message for the GetPublishedItemNames WSDL operation .

3.1.4.11.1.1 GetPublishedItemNamesSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetPublishedItemNames
```

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

3.1.4.11.1.2 GetPublishedItemNamesSoapOut

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

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

3.1.4.11.2 Elements

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

Element	Description
GetPublishedItemNames	The input data for the GetPublishedItemNames WSDL operation .
GetPublishedItemNamesResponse	The result data for the GetPublishedItemNames WSDL operation .

3.1.4.11.2.1 GetPublishedItemNames

The **GetPublishedItemNames** element specifies the input data for the **GetPublishedItemNames WSDL operation**.

```
<xs:element name="GetPublishedItemNames" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

sessionId: A **session identifier**. MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

3.1.4.11.2.2 GetPublishedItemNamesResponse

The **GetPublishedItemNamesResponse** element specifies the result data for the **GetPublishedItemNames WSDL operation**.

```
<xs:element name="GetPublishedItemNamesResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" maxOccurs="1" name="GetPublishedItemNamesResult"  
type="tns:ArrayOfWorkbookItem"/>  
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

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

GetPublishedItemNamesResult: An array of **WorkbookItem** elements as specified in section [3.1.4.11.3.1](#). MUST be present.

status: An array of **Status** elements as specified in section [2.2.4.2](#).

3.1.4.11.3 Complex Types

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

Complex type	Description
ArrayOfWorkbookItem	An array of WorkbookItem elements.
WorkbookItem	Specifies the name and type of a published item .

3.1.4.11.3.1 ArrayOfWorkbookItem

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

An array of **WorkbookItem** elements defined as follows.

```
<xs:complexType name="ArrayOfWorkbookItem" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="WorkbookItem" nillable="true"
type="tns:WorkbookItem"/>
  </xs:sequence>
</xs:complexType>
```

WorkbookItem: A **WorkbookItem** as specified in section [3.1.4.11.3.2](#).

3.1.4.11.3.2 WorkbookItem

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

Specifies the name and type of a **published item** and is defined as follows.

```
<xs:complexType name="WorkbookItem" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ItemType"/>
  </xs:sequence>
</xs:complexType>
```

Name: The name of the published item. MUST be present. MUST NOT be empty.

Type: The type of the published item as specified in section [3.1.4.11.4.1](#).

3.1.4.11.4 Simple Types

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

Simple type	Description
ItemType	Specifies the type of a published item .

3.1.4.11.4.1 ItemType

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

Specifies the type of a **published item** and is defined as follows.

```
<xs:simpleType name="ItemType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Chart"/>
    <xs:enumeration value="PivotTable"/>
    <xs:enumeration value="Table"/>
    <xs:enumeration value="NamedRange"/>
  </xs:restriction>
</xs:simpleType>
```

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

Value	Meaning
Chart	Specifies that the published item is a chart.
PivotTable	Specifies that the published item is a PivotTable .
Table	Specifies that the published item is a list .
NamedRange	Specifies that the published item is a named range .

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 GetRange

This operation is used to retrieve the contents of a specific **range** in the **workbook**.

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

```
<wsdl:operation name="GetRange" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetRangeSoapIn"/>
  <wsdl:output message="tns:GetRangeSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetRangeSoapIn** request message, and the protocol server responds with a **GetRangeSoapOut** response message, as follows:

1. The **rangeCoordinates** on the **sheet (2)** specified by **sheetName**, specify the location of the sheet (2) cells to retrieve.
2. Each element inside the **GetRangeResult** element MUST be of type **ArrayOfAnyType**. Each element inside **GetRangeResult** is a **row** of sheet (1) **cells** returned from the protocol server. The number of rows returned MUST be equal to the number of rows requested in the **rangeCoordinates** element.
3. Each element inside each of the **ArrayOfAnyType** type nested inside **GetRangeResult** MUST be a single workbook sheet (2) cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of **columns** requested in the **rangeCoordinates** element.
4. If **formatted** is false, all sheet (2) cells returned as part of the **GetRangeResult** element MUST either be empty and have the `xsi:nil` attribute set to true or be one of the following types: **Boolean**, **double**, **string**, or **CellError** (section [2.2.5.1](#)).
5. If **formatted** is true, all sheet cells returned as part of the **GetRangeResult** element MUST be type **string**, and MUST be number formatted according to how the sheet (1) cell is number formatted in the workbook.
6. If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.12.1 Messages

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

Message	Description
GetRangeSoapIn	The request WSDL message for the GetRange WSDL operation .

Message	Description
GetRangeSoapOut	The response WSDL message for the GetRange WSDL operation.

3.1.4.12.1.1 GetRangeSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetRange
```

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

3.1.4.12.1.2 GetRangeSoapOut

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

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

3.1.4.12.2 Elements

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

Element	Description
GetRange	The input data for the GetRange WSDL operation .
GetRangeResponse	The result data for the GetRange WSDL operation.

3.1.4.12.2.1 GetRange

The **GetRange** element specifies the input data for the **GetRange WSDL operation**.

```
<xs:element name="GetRange" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** that contains the **range**. See section [2.2.5.4](#) for specification. MUST be present. MUST NOT be empty.

rangeCoordinates: The range to retrieve.

- MUST be present.
- MUST NOT be empty, as specified in section [2.2.4.4](#).
- The product of **Height** and **Width** MUST NOT be greater than 1,000,000.

formatted: Specifies whether **GetRangeResult** will be number formatted.

3.1.4.12.2 GetRangeResponse

The **GetRangeResponse** element specifies the result data for the **GetRange WSDL operation**.

```
<xs:element name="GetRangeResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRangeResult"
type="tns:ArrayOfAnyType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRangeResult: The returned range value.

status: An array of status information as specified in section [2.2.4.2](#).

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 GetRangeA1

This operation is used to retrieve the contents of a **named range** in the **workbook**.

The following is the **WSDL** port type specification of the **GetRangeA1 WSDL operation**.

```
<wsdl:operation name="GetRangeA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetRangeA1SoapIn"/>
```

```
<wsdl:output message="tns:GetRangeA1SoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetRangeA1SoapIn** request message, and the protocol server responds with a **GetRangeA1SoapOut** response message, as follows:

- The **rangeName** specifies the named range of the cells on the **sheet (2)** to retrieve. The sheet (2) that contains the sheet (2) cells to retrieve is determined in the following manner:
- If **rangeName** specifies a sheet (2), that is the sheet (2) that MUST be used.
- Otherwise, the sheet (2) specified in **sheetName** MUST be used, as specified in section [2.2.5.4](#). If neither **sheetName** nor **rangeName** specify a sheet (2), the protocol server MAY [<7>](#) use any sheet (2) in the workbook.
- Each element inside the **GetRangeA1Result** element MUST be of type **ArrayOfAnyType**. Each element inside the **GetRangeA1Result** is a **row** of sheet (1) **cells** returned from the protocol server. The number of rows MUST be equal to the rows in the named range specified by **rangeName**.
- Each element inside each of the **ArrayOfAnyType** type nested inside **GetRangeA1Result** MUST be a single workbook sheet (2) cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of **columns** in the named range specified by **rangeName**.
- If **formatted** is false, all sheet (2) cells returned as part of the **GetRangeA1Result** element MUST either be empty and have the `xsi:nil` attribute set to true or be one of the following types: **Boolean**, **double**, **string**, or **CellError** (section 2.2.5.1).
- If **formatted** is true, all sheet (2) cells returned as part of the **GetRangeA1Result** element MUST be type **string**, and MUST be number formatted according to how the sheet (2) cell is number formatted in the workbook.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.13.1 Messages

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

Message	Description
GetRangeA1SoapIn	The request WSDL message for the GetRangeA1 WSDL operation .
GetRangeA1SoapOut	The response WSDL message for the GetRangeA1 WSDL operation .

3.1.4.13.1.1 GetRangeA1SoapIn

The request **WSDL message** for the **GetRangeA1 WSDL operation**.

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetRangeA1
```

The **SOAP body** contains the **GetRangeA1** element.

3.1.4.13.1.2 GetRangeA1SoapOut

The response **WSDL message** for the **GetRangeA1 WSDL operation**.

The **SOAP body** contains the **GetRangeA1Response** element.

3.1.4.13.2 Elements

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

Element	Description
GetRangeA1	The input data for the GetRangeA1 WSDL operation .
GetRangeA1Response	The result data for the GetRangeA1 WSDL operation .

3.1.4.13.2.1 GetRangeA1

The **GetRangeA1** element specifies the input data for the **GetRangeA1 WSDL operation**.

```
<xs:element name="GetRangeA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (1)** on which the **rangeName** is defined. See section [2.2.5.4](#) for specification. MUST be present.

rangeName: The name of the **range** as specified in section [2.2.5.2](#). MUST be present. MUST NOT be longer than 255 characters.

formatted: Specifies whether **GetRangeA1Result** will be number formatted.

3.1.4.13.2.2 GetRangeA1Response

The **GetRangeA1Response** element specifies the result data for the **GetRangeA1 WSDL operation**.

```
<xs:element name="GetRangeA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRangeA1Result"
type="tns:ArrayOfAnyType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetRangeA1Result: The returned range value.

status: An array of status information as specified in section [2.2.4.2](#).

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 GetSessionInformation

This operation retrieves information about a **session (2)** on the protocol server.

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

```
<wsdl:operation name="GetSessionInformation" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetSessionInformationSoapIn"/>
  <wsdl:output message="tns:GetSessionInformationSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetSessionInformationSoapIn** request message, and the protocol server responds with a **GetSessionInformationSoapOut** response message, as follows:

- The protocol server MUST return the **UI culture** and the **data culture** for the specified session (2). These are the languages specified in the [OpenWorkbook](#) operation or the languages that the protocol server fell back to, as specified in [OpenWorkbook](#).
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

3.1.4.14.1 Messages

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

Message	Description
GetSessionInformationSoapIn	The request WSDL message for the GetSessionInformation WSDL operation .
GetSessionInformationSoapOut	The response WSDL message for the GetSessionInformation WSDL operation .

3.1.4.14.1.1 GetSessionInformationSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformation
```

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

3.1.4.14.1.2 GetSessionInformationSoapOut

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

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

3.1.4.14.2 Elements

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

Element	Description
GetSessionInformation	The input data for the GetSessionInformation WSDL operation .

Element	Description
GetSessionInformationResponse	The result data for the GetSessionInformation WSDL operation.

3.1.4.14.2.1 GetSessionInformation

The **GetSessionInformation** element specifies the input data for the **GetSessionInformation** WSDL operation.

```
<xs:element name="GetSessionInformation" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session identifier**. MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

3.1.4.14.2.2 GetSessionInformationResponse

The **GetSessionInformationResponse** element specifies the result data for the **GetSessionInformation** WSDL operation.

```
<xs:element name="GetSessionInformationResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="serverVersion" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureNameUsed" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureNameUsed" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

serverVersion: The version of the server. MUST be present.

uiCultureNameUsed: The **UI culture**. MUST be present. MUST be a language tag, as specified in [\[RFC1766\]](#) section 2.

dataCultureNameUsed: The **data culture**. MUST be present. MUST be a language tag, as specified in [\[RFC1766\]](#) section 2.

status: An array of **Status** elements as specified in section [2.2.4.2](#).

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 GetSheetNames

The operation returns information about the **sheets** in a **workbook**.

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

```
<wsdl:operation name="GetSheetNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetSheetNamesSoapIn"/>
  <wsdl:output message="tns:GetSheetNamesSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetSheetNamesSoapIn** request message, and the protocol server responds with a **GetSheetNamesSoapOut** response message, as follows:

- If the **session (2)** is a **view session** in **view-only mode**, the number of elements inside **GetSheetNamesResult** MUST be equal to the number of **visible published sheets** in the workbook.
- If the session (2) is not a view session in view-only mode, the number of elements inside **GetSheetNamesResult** MUST be equal to the number of loaded sheets (1) in the workbook.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.15.1 Messages

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

Message	Description
GetSheetNamesSoapIn	The request WSDL message for the GetSheetNames WSDL operation .
GetSheetNamesSoapOut	The response WSDL message for the GetSheetNames WSDL operation .

3.1.4.15.1.1 GetSheetNamesSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames
```


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

3.1.4.15.1.2 GetSheetNamesSoapOut

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

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

3.1.4.15.2 Elements

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

Element	Description
GetSheetNames	The input data for the GetSheetNames WSDL operation .
GetSheetNamesResponse	The result data for the GetSheetNames WSDL operation .

3.1.4.15.2.1 GetSheetNames

The **GetSheetNames** element specifies the input data for the **GetSheetNames WSDL operation**.

```
<xs:element name="GetSheetNames" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

sessionId: A session identifier (2). MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

3.1.4.15.2.2 GetSheetNamesResponse

The **GetSheetNamesResponse** element specifies the result data for the **GetSheetNames WSDL operation**.

```
<xs:element name="GetSheetNamesResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element minOccurs="0" maxOccurs="1" name="GetSheetNamesResult"  
type="tns:ArrayOfSheetInfo"/>  
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

GetSheetNamesResult: An array of **SheetInfo** elements as specified in section [3.1.4.15.3.1](#). MUST be present.

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.15.3 Complex Types

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

Complex type	Description
ArrayOfSheetInfo	An array of SheetInfo elements.
SheetInfo	Specifies the name , type , and visible condition of a sheet (1) .

3.1.4.15.3.1 ArrayOfSheetInfo

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

An array of **SheetInfo** elements defined as follows.

```
<xs:complexType name="ArrayOfSheetInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="SheetInfo" nillable="true"
type="tns:SheetInfo"/>
  </xs:sequence>
</xs:complexType>
```

SheetInfo: A **SheetInfo** as specified in section [3.1.4.15.3.2](#).

3.1.4.15.3.2 SheetInfo

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

Specifies the **name**, **type**, and **visible** condition of a **sheet (1)** and is defined as follows.

```
<xs:complexType name="SheetInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:SheetType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Visibility" type="tns:SheetVisibility"/>
  </xs:sequence>
</xs:complexType>
```

Name: The name of the sheet (1). **MUST** be present.

Type: The type of the sheet (1) as specified in section [3.1.4.15.4.1](#).

Visibility: The visible condition of the sheet (1) as specified in section [3.1.4.15.4.2](#).

3.1.4.15.4 Simple Types

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

Simple type	Description
SheetType	Specifies the type of a sheet (1) .
SheetVisibility	Specifies the visible condition of a sheet (1).

3.1.4.15.4.1 SheetType

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

Specifies the type of a **sheet (1)** and is defined as follows.

```
<xs:simpleType name="SheetType" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="Worksheet"/>  
    <xs:enumeration value="Chartsheet"/>  
  </xs:restriction>  
</xs:simpleType>
```

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

Value	Meaning
Worksheet	A worksheet .
Chartsheet	A chart sheet .

3.1.4.15.4.2 SheetVisibility

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

Specifies the **visible** condition of a **sheet (1)** and is defined as follows.

```
<xs:simpleType name="SheetVisibility" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="Visible"/>  
    <xs:enumeration value="Hidden"/>  
    <xs:enumeration value="VeryHidden"/>  
  </xs:restriction>  
</xs:simpleType>
```

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

Value	Meaning
Visible	Specifies that the sheet (1) is visible.
Hidden	Specifies that the sheet (1) is not visible, but can be unhidden via the client UI.
VeryHidden	Specifies that the sheet (1) is not visible and cannot be unhidden via the client UI.

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 GetWorkbook

This operation is used to retrieve a **workbook** file.

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

```
<wsdl:operation name="GetWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetWorkbookSoapIn"/>
  <wsdl:output message="tns:GetWorkbookSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetWorkbookSoapIn** request message, and the protocol server responds with a **GetWorkbookSoapOut** response message, as follows:

- If **workbookType** is **FullWorkbook** or **FullSnapshot** and the user does not have **Open Item permission**, the protocol server returns a SOAP fault with subcode **EcsSnapshotAccessDenied**.
- If the protocol server cannot retrieve the **workbook file** in the form specified by **workbookType**, the protocol server returns a SOAP fault with subcode **SaveFailed**.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
EcsSnapshotAccessDenied
SaveFailed

3.1.4.16.1 Messages

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

Message	Description
GetWorkbookSoapIn	The request WSDL message for the GetWorkbook WSDL operation .
GetWorkbookSoapOut	The response WSDL message for the GetWorkbook WSDL operation .

3.1.4.16.1.1 GetWorkbookSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook
```

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

3.1.4.16.1.2 GetWorkbookSoapOut

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

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

3.1.4.16.2 Elements

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

Element	Description
GetWorkbook	The input data for the GetWorkbook WSDL operation .
GetWorkbookResponse	The result data for the GetWorkbook WSDL operation .

3.1.4.16.2.1 GetWorkbook

The **GetWorkbook** element specifies the input data for the **GetWorkbook WSDL operation**.

```
<xs:element name="GetWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType" type="tns:WorkbookType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session (2)** identifier. MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

workbookType: The form of the requested workbook as specified in section [2.2.5.5](#).

3.1.4.16.2.2 GetWorkbookResponse

The **GetWorkbookResponse** element specifies the result data for the **GetWorkbook WSDL operation**.

```
<xs:element name="GetWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookResult"
type="xs:base64Binary"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetWorkbookResult: A **workbook** file. MUST be present.

status: An array of status information as specified in section [2.2.4.2](#).

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 GetWorkbookConnections

Returns a list of **data connection** names in the workbook.

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

```
<wsdl:operation name="GetWorkbookConnections" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetWorkbookConnectionsSoapIn"/>
  <wsdl:output message="tns:GetWorkbookConnectionsSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **GetWorkbookConnectionsSoapIn** request **WSDL message**, and the protocol server responds with a **GetWorkbookConnectionsSoapOut** response WSDL message, as follows:

- The protocol server returns a list of the data connection names.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.17.1 Messages

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

Message	Description
GetWorkbookConnectionsSoapIn	The request WSDL message for the GetWorkbookConnections WSDL operation .
GetWorkbookConnectionsSoapOut	The response WSDL message for the GetWorkbookConnections WSDL operation .

3.1.4.17.1.1 GetWorkbookConnectionsSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbookConnections
```

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

3.1.4.17.1.2 GetWorkbookConnectionsSoapOut

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

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

3.1.4.17.2 Elements

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

Element	Description
GetWorkbookConnections	The input data for the GetWorkbookConnections WSDL operation .
GetWorkbookConnectionsResponse	The result data for the GetWorkbookConnections WSDL operation .

3.1.4.17.2.1 GetWorkbookConnections

The **GetWorkbookConnections** element specifies the input data for the **GetWorkbookConnections WSDL operation**.

```
<xs:element name="GetWorkbookConnections" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session (2)** identifier. MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

3.1.4.17.2.2 GetWorkbookConnectionsResponse

The **GetWorkbookConnectionsResponse** element specifies the result data for the **GetWorkbookConnections WSDL operation**.

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

```

    <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookConnectionsResult"
type="tns:ArrayOfString"/>
    <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

GetWorkbookConnectionsResult: An array of **data connection** names.

status: An array of status information as specified in section [2.2.4.2](#).

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 NewWorkbook

This operation is used to create a **new workbook session**.

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

```

<wsdl:operation name="NewWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:NewWorkbookSoapIn"/>
  <wsdl:output message="tns:NewWorkbookSoapOut"/>
</wsdl:operation>

```

If the language represented by **uiCultureName** is unsupported on the protocol server, the protocol server MAY [<8>](#) fall back to some other supported language. **uiCultureName** MAY [<9>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

If the language represented by **dataCultureName** is unsupported on the protocol server, the protocol server MAY [<10>](#) fall back to some other supported language. **dataCultureName** MAY [<11>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

If the operation succeeds, the protocol server MUST return the **session identifier** of the newly created **session (2)** in the response message, to be used in any future operations in the newly created session.

If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.18.1 Messages

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

Message	Description
NewWorkbookSoapIn	The request WSDL message for the NewWorkbook WSDL operation .
NewWorkbookSoapOut	The response WSDL message for the NewWorkbook WSDL operation .

3.1.4.18.1.1 NewWorkbookSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/NewWorkbook
```

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

3.1.4.18.1.2 NewWorkbookSoapOut

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

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

3.1.4.18.2 Elements

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

Element	Description
NewWorkbook	The input data for the NewWorkbook WSDL operation .
NewWorkbookResponse	The result data for the NewWorkbook WSDL operation .

3.1.4.18.2.1 NewWorkbook

The **NewWorkbook** element specifies the input data for the **NewWorkbook WSDL operation**.

```
<xs:element name="NewWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="path" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

</xs:element>

path: The **trusted location** of the **new workbook session**. MUST be present. MUST NOT be empty. MUST NOT be longer than 4095 characters. MUST be a **UNC**, or a **URI** as specified in [RFC2396](#).

uiCultureName: A language tag, as specified in [RFC1766](#), Section 2. Specifies the **UI culture**. MUST NOT be longer than 4095 characters.

dataCultureName: A language tag, as specified in [RFC1766](#), Section 2. Specifies the **data culture**. MUST NOT be longer than 4095 characters.

3.1.4.18.2.2 NewWorkbookResponse

The **NewWorkbookResponse** element specifies the result data for the **NewWorkbook WSDL operation**.

```
<xs:element name="NewWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="NewWorkbookResult" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

NewWorkbookResult: The **session identifier** of the newly created **session (2)**. MUST be present.

status: The status information returned by the protocol server.

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 OpenWorkbook

This operation is used to create a new **view session**, giving access to a workbook on the protocol server.

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

```

<wsdl:operation name="OpenWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:OpenWorkbookSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookSoapOut"/>
</wsdl:operation>

```

The protocol client sends an **OpenWorkbookSoapIn** request message, and the protocol server responds with an **OpenWorkbookSoapOut** response message, as follows:

1. The workbook file, specified by **workbookPath**, is specified as follows:
 1. It SHOULD be in a format specified by [\[MS-XLSB\]](#), [\[ISO/IEC29500-1:2011\]](#), [\[MS-OFFMACRO2\]](#), or [\[ODF\]](#), but MAY be in any other format.
 2. If it is in the [\[ISO/IEC29500-1:2011\]](#) format:
 1. It MAY [<12>](#) contain any valid SpreadsheetML records in that format.
 2. If a **odcFile** attribute of a **connection** element ([\[ISO/IEC29500-1:2011\]](#), Part 1, Section 18.13.1) is present, it SHOULD be the path of a file in the format specified by [\[MS-ODCFF\]](#).
 3. If a **connection** attribute of **dbPr** element ([\[ISO/IEC29500-1:2011\]](#), Part 1, Section 18.13.3) is present, it SHOULD be in the format specified by [\[MS-ODBCSTR\]](#) or [\[MS-OLEDBSTR\]](#).
 3. If it is in the [\[MS-XLSB\]](#) format:
 1. It MAY [<13>](#) contain any valid records in that format.
 2. If a **brtBeginExternalConnection** record is present and it has the **stConnectionFile** field present, the **stConnectionFile** SHOULD be the path of a file in the format specified by [\[MS-ODCFF\]](#).
 3. If a **BrtBeginECDBProps** record is present, its **stConn** field SHOULD be in the format specified by [\[MS-ODBCSTR\]](#) or [\[MS-OLEDBSTR\]](#).
 4. If a file in [\[MS-ODCFF\]](#) is referenced by the workbook file:
 1. **SourceFile** element of **OfficeDataConnection** ([\[MS-ODCFF\]](#) section 2.7.1.1) MUST NOT be present.
 2. **Parameter** elements of **CT_Connection** ([\[MS-ODCFF\]](#) section 2.7.1.1) MUST be ignored.
2. If the language represented by **uiCultureName** is unsupported on the protocol server, the protocol server MAY [<14>](#) fall back to some other supported language. **uiCultureName** MAY [<15>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.
3. If the language represented by **dataCultureName** is unsupported on the protocol server, the protocol server MAY [<16>](#) fall back to some other supported language. **dataCultureName** MAY [<17>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.
4. If the operation succeeds, the protocol server MUST return the **session identifier** of the newly created **session (2)** in the response message, to be used in any future operations in the newly created session (2).
5. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FileOpenNotFound

Subcode
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError
WorkbookNotSupported

3.1.4.19.1 Messages

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

Message	Description
OpenWorkbookSoapIn	The request WSDL message for the OpenWorkbook WSDL operation .
OpenWorkbookSoapOut	The response WSDL message for the OpenWorkbook WSDL operation .

3.1.4.19.1.1 OpenWorkbookSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook
```

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

3.1.4.19.1.2 OpenWorkbookSoapOut

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

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

3.1.4.19.2 Elements

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

Element	Description
OpenWorkbook	The input data for the OpenWorkbook WSDL operation .
OpenWorkbookResponse	The result data for the OpenWorkbook WSDL operation .

3.1.4.19.2.1 OpenWorkbook

The **OpenWorkbook** element specifies the input data for the **OpenWorkbook WSDL operation**.

```
<xs:element name="OpenWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

workbookPath: The location of the workbook file. MUST be present. MUST NOT be empty. MUST NOT be longer than 4095 characters. MUST be a **URI** as specified in [\[RFC2396\]](#), or a **UNC**.

uiCultureName: A language tag, as specified in [\[RFC1766\]](#), Section 2. Specifies the UI culture. MUST NOT be longer than 4095 characters.

dataCultureName: A language tag, as specified in [\[RFC1766\]](#), Section 2. Specifies the data culture. MUST NOT be longer than 4095 characters.

3.1.4.19.2 OpenWorkbookResponse

The **OpenWorkbookResponse** element specifies the result data for the **OpenWorkbook WSDL operation**.

```
<xs:element name="OpenWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookResult" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

OpenWorkbookResult: The **session identifier** of the newly created **session (2)**. MUST be present.

status: The status information returned by the protocol server.

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 OpenWorkbookEx

This operation is used to create a new **view session**, giving access to a workbook on the protocol server.

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

```
<wsdl:operation name="OpenWorkbookEx" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:OpenWorkbookExSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookExSoapOut"/>
</wsdl:operation>
```

For details, see the **OpenWorkbook** operation (section [3.1.4.19](#)). When processing the **OpenWorkbookEx** operation the protocol server MUST suppress **connected data refresh**.

3.1.4.20.1 Messages

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

Message	Description
OpenWorkbookExSoapIn	The request WSDL message for the OpenWorkbookEx WSDL operation .
OpenWorkbookExSoapOut	The response WSDL message for the OpenWorkbookEx WSDL operation .

3.1.4.20.1.1 OpenWorkbookExSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookEx
```

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

3.1.4.20.1.2 OpenWorkbookExSoapOut

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

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

3.1.4.20.2 Elements

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

Element	Description
OpenWorkbookEx	The input data for the OpenWorkbookEx WSDL operation .
OpenWorkbookExResponse	The result data for the OpenWorkbookEx WSDL operation .

3.1.4.20.2.1 OpenWorkbookEx

The **OpenWorkbookEx** element specifies the input data for the **OpenWorkbookEx WSDL operation**.

```
<xs:element name="OpenWorkbookEx" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="exclusive" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

workbookPath: See section [3.1.4.19.2.1](#).

uiCultureName: See section 3.1.4.19.2.1.

dataCultureName: See section 3.1.4.19.2.1.

exclusive: If true, the data model in the new **view session** MUST NOT be shared with any other **sessions**, and the database where the model will be loaded (as specified in section [3.1.4.6.3.1](#)) MUST NOT be read-only.

3.1.4.20.2.2 OpenWorkbookExResponse

The **OpenWorkbookExResponse** element specifies the result data for the **OpenWorkbookEx WSDL operation**.

```
<xs:element name="OpenWorkbookExResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookExResult" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

OpenWorkbookExResult: See section [3.1.4.19.2.2](#).

status: See section 3.1.4.19.2.2.

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 OpenWorkbookForEditing

This operation is used to create a new **edit session**, giving access to a **workbook** on the protocol server.

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

```
<wsdl:operation name="OpenWorkbookForEditing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:OpenWorkbookForEditingSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookForEditingSoapOut"/>
</wsdl:operation>
```

The protocol client sends an **OpenWorkbookForEditingSoapIn** request message, and the protocol server responds with an **OpenWorkbookForEditingSoapOut** response message, as follows:

1. The workbook file, specified by **workbookPath**, is specified as follows:
 1. It SHOULD be in a format specified by [\[MS-XLSB\]](#), [\[ISO/IEC29500-1:2011\]](#), [\[MS-OFFMACRO2\]](#), or [\[ODF\]](#), but MAY be in any other format.
 2. If it is in the [ISO/IEC29500-1:2011] format:
 1. It MAY [<18>](#) contain any valid SpreadsheetML records in that format.
 2. If an **odcFile** attribute of a **connection** element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.1) is present, it SHOULD be the path of a file in the format specified by [\[MS-ODCFF\]](#).
 3. If a **connection** attribute of **dbPr** element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.3) is present, it SHOULD be in the format specified by [\[MS-ODBCSTR\]](#) or [\[MS-OLEDBSTR\]](#).
 3. If it is in the [MS-XLSB] format:
 1. It MAY [<19>](#) contain any valid records in that format.
 2. If a **brtBeginExternalConnection** record is present and it has the **stConnectionFile** field present, the **stConnectionFile** SHOULD be the path of a file in the format specified by [MS-ODCFF].
 3. If a **BrtBeginECdbProps** record is present, its **stConn** field SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].
 4. If a file in [MS-ODCFF] is referenced by the workbook file:
 1. **SourceFile** element of **OfficeDataConnection** ([MS-ODCFF] section 2.7.1.1) MUST NOT be present.
 2. **Parameter** elements of **CT_Connection** ([MS-ODCFF] section 2.2.1) MUST be ignored.
2. If the language represented by **uiCultureName** is unsupported on the protocol server, the protocol server MAY [<20>](#) fall back to some other supported language. **uiCultureName** MAY [<21>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.
3. If the language represented by **dataCultureName** is unsupported on the protocol server, the protocol server MAY [<22>](#) fall back to some other supported language. **dataCultureName** MAY [<23>](#) be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

4. If the operation succeeds, the protocol server MUST return the **session identifier** of the newly created **session (2)** in the response message, to be used in any future operations in the newly created session (2).
6. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FileOpenNotFound
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError

3.1.4.21.1 Messages

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

Message	Description
OpenWorkbookForEditingSoapIn	The request WSDL message for the OpenWorkbookForEditing WSDL operation .
OpenWorkbookForEditingSoapOut	The response WSDL message for the OpenWorkbookForEditing WSDL operation .

3.1.4.21.1.1 OpenWorkbookForEditingSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookForEditing
```

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

3.1.4.21.1.2 OpenWorkbookForEditingSoapOut

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

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

3.1.4.21.2 Elements

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

Element	Description
OpenWorkbookForEditing	The input data for the OpenWorkbookForEditing WSDL operation .
OpenWorkbookForEditingResponse	The result data for the OpenWorkbookForEditing WSDL operation.

3.1.4.21.2.1 OpenWorkbookForEditing

The **OpenWorkbookForEditing** element specifies the input data for the **OpenWorkbookForEditing WSDL operation**.

```
<xs:element name="OpenWorkbookForEditing" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

workbookPath: The location of the **workbook** file. MUST be present. MUST NOT be empty. MUST NOT be longer than 4095 characters. MUST be a **URI** as specified in [\[RFC2396\]](#), or a **UNC**.

uiCultureName: A language tag, as specified in [\[RFC1766\]](#), Section 2. Specifies the **UI culture**. MUST NOT be longer than 4095 characters.

dataCultureName: A language tag, as specified in [\[RFC1766\]](#), Section 2. Specifies the **data culture**. MUST NOT be longer than 4095 characters.

3.1.4.21.2.2 OpenWorkbookForEditingResponse

The **OpenWorkbookForEditingResponse** element specifies the result data for the **OpenWorkbookForEditing WSDL operation**.

```
<xs:element name="OpenWorkbookForEditingResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookForEditingResult"
        type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

OpenWorkbookForEditingResult: The **session identifier** of the newly created **session (2)**. MUST be present.

status: An array of status information as specified in section [2.2.4.2](#).

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 Refresh

This operation is used to **refresh connected data** on a **workbook**.

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

```
<wsdl:operation name="Refresh" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:RefreshSoapIn"/>
  <wsdl:output message="tns:RefreshSoapOut"/>
</wsdl:operation>
```

The protocol client sends a **RefreshSoapIn** request message, and the protocol server responds with a **RefreshSoapOut** response message, as follows:

1. If **connectionName** is not present or is empty, the protocol server refreshes all **data connections** in the workbook if any exist. If **connectionName** is present and is not empty, the protocol server refreshes the specified data connections in the workbook.
2. If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.22.1 Messages

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

Message	Description
RefreshSoapIn	The request WSDL message for the Refresh WSDL operation .
RefreshSoapOut	The response WSDL message for the Refresh WSDL operation .

3.1.4.22.1.1 RefreshSoapIn

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

The **SOAP action** value is:

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

3.1.4.22.1.2 RefreshSoapOut

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

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

3.1.4.22.2 Elements

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

Element	Description
Refresh	The input data for the Refresh WSDL operation .
RefreshResponse	The result data for the Refresh WSDL operation .

3.1.4.22.2.1 Refresh

The **Refresh** element specifies the input data for the **Refresh WSDL operation**.

```
<xs:element name="Refresh" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

connectionName: A name of a **data connection** in the **workbook**. MUST NOT be longer than 255 characters.

3.1.4.22.2.2 RefreshResponse

The **RefreshResponse** element specifies the result data for the **Refresh WSDL operation**.

```
<xs:element name="RefreshResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of **Status** elements as specified in section [2.2.4.2](#).

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 RefreshDetached

MUST NOT be used.

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

```
<wsdl:operation name="RefreshDetached" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:RefreshDetachedSoapIn"/>
  <wsdl:output message="tns:RefreshDetachedSoapOut"/>
</wsdl:operation>
```

The **RefreshDetachedSoapIn** and **RefreshDetachedSoapOut** messages MUST NOT be used.

3.1.4.23.1 Messages

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

Message	Description
RefreshDetachedSoapIn	The request WSDL message for the RefreshDetached WSDL operation .
RefreshDetachedSoapOut	The response WSDL message for the RefreshDetached WSDL operation .

3.1.4.23.1.1 RefreshDetachedSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/RefreshDetached
```

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

3.1.4.23.1.2 RefreshDetachedSoapOut

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

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

3.1.4.23.2 Elements

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

Element	Description
RefreshDetached	The input data for the RefreshDetached WSDL operation .
RefreshDetachedResponse	The result data for the RefreshDetached WSDL operation .

3.1.4.23.2.1 RefreshDetached

The **RefreshDetached** element specifies the input data for the **RefreshDetached WSDL operation**.

```
<xs:element name="RefreshDetached" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionDetails"
type="tns:ArrayOfConnectionDetails"/>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: MUST NOT be used.

connectionDetails: MUST NOT be used.

requestId: MUST NOT be used.

3.1.4.23.2.2 RefreshDetachedResponse

The **RefreshDetachedResponse** element specifies the result data for the **RefreshDetached WSDL operation**.

```
<xs:element name="RefreshDetachedResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
      <xs:element minOccurs="0" maxOccurs="1" name="detailedRefreshResult"
type="tns:ArrayOfDetailedRefreshResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

requestId: MUST NOT be used.

status: MUST NOT be used.

detailedRefreshResult: MUST NOT be used.

3.1.4.23.3 Complex Types

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

Complex type	Description
ArrayOfConnectionDetails	MUST NOT be used.
ArrayOfDetailedRefreshResult	MUST NOT be used.
ConnectionDetails	MUST NOT be used.
DetailedRefreshResult	MUST NOT be used.

3.1.4.23.3.1 ArrayOfConnectionDetails

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

```
<xs:complexType name="ArrayOfConnectionDetails" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ConnectionDetails" nillable="true"
type="tns:ConnectionDetails"/>
  </xs:sequence>
</xs:complexType>
```

ConnectionDetails: MUST NOT be used.

3.1.4.23.3.2 ArrayOfDetailedRefreshResult

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

MUST NOT be used.

```
<xs:complexType name="ArrayOfDetailedRefreshResult"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="DetailedRefreshResult"
nillable="true" type="tns:DetailedRefreshResult"/>
  </xs:sequence>
</xs:complexType>
```

DetailedRefreshResult: MUST NOT be used.

3.1.4.23.3.3 ConnectionDetails

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

MUST NOT be used.

```
<xs:complexType name="ConnectionDetails" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConnectionName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TargetApplicationId" type="xs:string"/>
  </xs:sequence>
```

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

ConnectionName: MUST NOT be used.

TargetApplicationId: MUST NOT be used.

3.1.4.23.3.4 DetailedRefreshResult

Namespace: http://schemas.microsoft.com/office/excel/server/webservices

MUST NOT be used.

```
<xs:complexType name="DetailedRefreshResult" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConnectionName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ErrorMessage" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

ConnectionName: MUST NOT be used.

ErrorMessage: MUST NOT be used.

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 RefreshEx

This operation is used to **refresh connected data** on a **workbook**.

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

```
<wsdl:operation name="RefreshEx" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:RefreshExSoapIn"/>
  <wsdl:output message="tns:RefreshExSoapOut"/>
</wsdl:operation>
```

For details, see the **Refresh** operation (section [3.1.4.22](#)).

3.1.4.24.1 Messages

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

Message	Description
RefreshExSoapIn	The request WSDL message for the RefreshEx WSDL operation .
RefreshExSoapOut	The response WSDL message for the RefreshEx WSDL operation .

3.1.4.24.1.1 RefreshExSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/RefreshEx
```

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

3.1.4.24.1.2 RefreshExSoapOut

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

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

3.1.4.24.2 Elements

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

Element	Description
RefreshEx	The input data for the RefreshEx WSDL operation .
RefreshExResponse	The result data for the RefreshEx WSDL operation .

3.1.4.24.2.1 RefreshEx

The **RefreshEx** element specifies the input data for the **RefreshEx WSDL operation**.

```
<xs:element name="RefreshEx" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="targetApplicationId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

connectionName: A name of a **data connection** in the **workbook**. MUST NOT be longer than 255 characters.

targetApplicationId: Single Sign On identifier, as defined in [\[ISO/IEC29500-1:2012\]](#) section 18.13.1, used when refreshing data connections.

3.1.4.24.2 RefreshExResponse

The **RefreshExResponse** element specifies the result data for the **RefreshEx WSDL operation**.

```
<xs:element name="RefreshExResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of **Status** elements as specified in section [2.2.4.2](#).

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 SaveWorkbook

This operation is used to save a **workbook** associated with an open **session (2)** to the same location from which it was opened.

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

```
<wsdl:operation name="SaveWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SaveWorkbookSoapIn"/>
  <wsdl:output message="tns:SaveWorkbookSoapOut"/>
</wsdl:operation>
```

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

If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other

subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FileSaveGeneric
OperationNotSupportedInViewOnly
SaveFailed
SavingUnsupportedInViewMode
SavingWorkbookAccessDenied
SavingWorkbookLockedOrCheckedOut
SavingWorkbookNoWorkbookName

3.1.4.25.1 Messages

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

Message	Description
SaveWorkbookSoapIn	The request WSDL message for the SaveWorkbook WSDL operation .
SaveWorkbookSoapOut	The response WSDL message for the SaveWorkbook WSDL operation .

3.1.4.25.1.1 SaveWorkbookSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbook
```

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

3.1.4.25.1.2 SaveWorkbookSoapOut

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

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

3.1.4.25.2 Elements

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

Element	Description
SaveWorkbook	The input data for the SaveWorkbook WSDL operation .
SaveWorkbookResponse	The result data for the SaveWorkbook WSDL operation .

3.1.4.25.2.1 SaveWorkbook

The **SaveWorkbook** element specifies the input data for the **SaveWorkbook WSDL operation**.

```
<xs:element name="SaveWorkbook" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

3.1.4.25.2.2 SaveWorkbookResponse

The **SaveWorkbookResponse** element specifies the result data for the **SaveWorkbook WSDL operation**.

```
<xs:element name="SaveWorkbookResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

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 SaveWorkbookCopy

This operation is used to save a **workbook** associated with an open **session (2)** to a specified location, and in a specified format.

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

```
<wsdl:operation name="SaveWorkbookCopy" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SaveWorkbookCopySoapIn"/>
  <wsdl:output message="tns:SaveWorkbookCopySoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SaveWorkbookSoapIn** request message and the protocol server responds with a **SaveWorkbookSoapOut** response message, as follows:

- If the protocol server cannot save the workbook in the form specified by **workbookType**, the protocol server returns a SOAP fault with subcode **SaveFailed**.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
EcsSnapshotAccessDenied
FileSaveGeneric
SaveFailed
SavingWorkbookAccessDenied
SavingWorkbookFileNameExists
SavingWorkbookInvalidFileName
SavingWorkbookInvalidLocation

3.1.4.26.1 Messages

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

Message	Description
SaveWorkbookCopySoapIn	The request WSDL message for the SaveWorkbookCopy WSDL operation .
SaveWorkbookCopySoapOut	The response WSDL message for the SaveWorkbookCopy WSDL operation .

3.1.4.26.1.1 SaveWorkbookCopySoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopy
```

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

3.1.4.26.1.2 SaveWorkbookCopySoapOut

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

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

3.1.4.26.2 Elements

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

Element	Description
SaveWorkbookCopy	The input data for the SaveWorkbookCopy WSDL operation .
SaveWorkbookCopyResponse	The result data for the SaveWorkbookCopy WSDL operation .

3.1.4.26.2.1 SaveWorkbookCopy

The **SaveWorkbookCopy** element specifies the input data for the **SaveWorkbookCopy WSDL operation**.

```
<xs:element name="SaveWorkbookCopy" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType" type="tns:WorkbookType"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

workbookPath: The location of the **workbook** file to which to write.

- MUST be present.

If the specified value is not a **Uniform Resource Identifier (URI)** as specified in [\[RFC2396\]](#), or a **Universal Naming Convention (UNC)**, the protocol server MUST return a **SOAP fault** response with subcode **SavingWorkbookInvalidFileName**.

- The protocol server ignores the number sign ("#") and fragment, as specified in [\[RFC3986\]](#).

workbookType: A **WorkbookType** as specified in section [2.2.5.5](#).

saveOptions: A combination of 0 or more values of **SaveOptions** (section [2.2.5.3](#)). MUST be present.

3.1.4.26.2 SaveWorkbookCopyResponse

The **SaveWorkbookCopyResponse** element specifies the result data for the **SaveWorkbookCopy WSDL operation**.

```
<xs:element name="SaveWorkbookCopyResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

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 SaveWorkbookCopyDetached

MUST NOT be used.

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

```
<wsdl:operation name="SaveWorkbookCopyDetached"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SaveWorkbookCopyDetachedSoapIn"/>
  <wsdl:output message="tns:SaveWorkbookCopyDetachedSoapOut"/>
</wsdl:operation>
```

The **SaveWorkbookCopyDetachedSoapIn** and **SaveWorkbookCopyDetachedSoapOut** messages MUST NOT be used.

3.1.4.27.1 Messages

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

Message	Description
SaveWorkbookCopyDetachedSoapIn	The request WSDL message for the SaveWorkbookCopyDetached WSDL operation .
SaveWorkbookCopyDetachedSoapOut	The response WSDL message for the SaveWorkbookCopyDetached WSDL operation .

3.1.4.27.1.1 SaveWorkbookCopyDetachedSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopyDetached
```

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

3.1.4.27.1.2 SaveWorkbookCopyDetachedSoapOut

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

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

3.1.4.27.2 Elements

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

Element	Description
SaveWorkbookCopyDetached	The input data for the SaveWorkbookCopyDetached WSDL operation .
SaveWorkbookCopyDetachedResponse	The result data for the SaveWorkbookCopyDetached WSDL operation .

3.1.4.27.2.1 SaveWorkbookCopyDetached

The **SaveWorkbookCopyDetached** element specifies the input data for the **SaveWorkbookCopyDetached WSDL operation**.

```
<xs:element name="SaveWorkbookCopyDetached" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"/>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



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

sessionId: MUST NOT be used.

workbookPath: MUST NOT be used.

saveOptions: MUST NOT be used.

requestId: MUST NOT be used.

3.1.4.27.2 SaveWorkbookCopyDetachedResponse

The **SaveWorkbookCopyDetachedResponse** element specifies the result data for the **SaveWorkbookCopyDetached WSDL operation**.

```
<xs:element name="SaveWorkbookCopyDetachedResponse"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

requestId: MUST NOT be used.

status: MUST NOT be used.

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.

3.1.4.28 SetCalculationOptions

This operation is used to set the **workbook calculation mode**.

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

```

<wsdl:operation name="SetCalculationOptions" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetCalculationOptionsSoapIn"/>
  <wsdl:output message="tns:SetCalculationOptionsSoapOut"/>
</wsdl:operation>

```

The protocol client sends a **SetCalculationOptionsSoapIn** request message and the protocol server responds with a **SetCalculationOptionsSoapOut** response message as follows:

- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD be one of the subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

3.1.4.28.1 Messages

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

Message	Description
SetCalculationOptionsSoapIn	The request WSDL message for the SetCalculationOptions WSDL operation .
SetCalculationOptionsSoapOut	The response WSDL message for the SetCalculationOptions WSDL operation .

3.1.4.28.1.1 SetCalculationOptionsSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetCalculationOptions
```

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

3.1.4.28.1.2 SetCalculationOptionsSoapOut

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

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

3.1.4.28.2 Elements

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

Element	Description
SetCalculationOptions	The input data for the SetCalculationOptions WSDL operation .
SetCalculationOptionsResponse	The result data for the SetCalculationOptions WSDL operation .

3.1.4.28.2.1 SetCalculationOptions

The **SetCalculationOptions** element specifies the input data for the **SetCalculationOptions WSDL operation**.

```
<xs:element name="SetCalculationOptions" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="calcType"
type="tns:WorkbookCalculation"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

calcType: The **calculation mode** value as specified in **WorkbookCalculation**.

3.1.4.28.2.2 SetCalculationOptionsResponse

The **SetCalculationOptionsResponse** element specifies the result data for the **SetCalculationOptions WSDL operation**.

```
<xs:element name="SetCalculationOptionsResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.28.3 Complex Types

None.

3.1.4.28.4 Simple Types

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

Simple type	Description
WorkbookCalculation	The WorkbookCalculation type is an enumeration of a set of options that specify how the protocol server calculates the workbook .

3.1.4.28.4.1 WorkbookCalculation

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

The **WorkbookCalculation** type is an enumeration of a set of options that specify how the protocol server **calculates** the **workbook**. This type is defined as follows.

```
<xs:simpleType name="WorkbookCalculation" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Manual"/>
    <xs:enumeration value="Automatic"/>
    <xs:enumeration value="Semiautomatic"/>
  </xs:restriction>
</xs:simpleType>
```

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

Value	Meaning
Manual	Specifies workbook is in manual calculation mode .
Automatic	Specifies workbook is in automatic calculation mode .
Semiautomatic	Specifies workbook is in automatic calculation mode, except for data tables .

3.1.4.28.5 Attributes

None.

3.1.4.28.6 Groups

None.

3.1.4.28.7 Attribute Groups

None.

3.1.4.29 SetCell

This operation is used to set a value to a specific **worksheet cell** in the **workbook**.

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

```
<wsdl:operation name="SetCell" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetCellSoapIn"/>
  <wsdl:output message="tns:SetCellSoapOut"/>
</wsdl:operation>
```

1. The **row** and **column** fields on the **sheets (2)** specified by **sheetName** specify the location of the worksheet cell to set.
2. **cellValue** MUST either be empty and have the **xsi:nil** attribute set to **true** or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string**, or **unsigned byte**.
 1. If the value is of type **string**, it MUST NOT be longer than 32,767 characters.
 2. If the protocol server does not support the value it MUST return a **SOAP fault**.

3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.
3. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FormulaParseFailed
InvalidSheetName
CannotEditPivotTable

3.1.4.29.1 Messages

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

Message	Description
SetCellSoapIn	The request WSDL message for the SetCell WSDL operation .
SetCellSoapOut	The response WSDL message for the SetCell WSDL operation .

3.1.4.29.1.1 SetCellSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetCell
```

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

3.1.4.29.1.2 SetCellSoapOut

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

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

3.1.4.29.2 Elements

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

Element	Description
SetCell	The input data for the SetCell WSDL operation .

Element	Description
SetCellResponse	The result data for the SetCell WSDL operation.

3.1.4.29.2.1 SetCell

The **SetCell** element specifies the input data for the **SetCell WSDL operation**.

```
<xs:element name="SetCell" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** that contains the **worksheet cell** to set. See section [2.2.5.4](#). MUST be present. MUST NOT be empty.

row: The **row** index of the worksheet cell. MUST be greater than or equal to 0 and less than or equal to 1,048,575.

column: The **column** index of the worksheet cell. MUST be greater than or equal to 0 and less than or equal to 16,383.

cellValue: The value to set in the worksheet cell.

3.1.4.29.2.2 SetCellResponse

The **SetCellResponse** element specifies the result data for the **SetCell WSDL operation**.

```
<xs:element name="SetCellResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.29.3 Complex Types

None.

3.1.4.29.4 Simple Types

None.

3.1.4.29.5 Attributes

None.

3.1.4.29.6 Groups

None.

3.1.4.29.7 Attribute Groups

None.

3.1.4.30 SetCellA1

This operation is used to set a value to a specific **worksheet cell** in the **workbook**.

The following is the **WSDL** port type specification of the **SetCellA1 WSDL operation**.

```
<wsdl:operation name="SetCellA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetCellA1SoapIn"/>
  <wsdl:output message="tns:SetCellA1SoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SetCellA1SoapIn** request message, and the protocol server responds with a **SetCellA1SoapOut** response message, as follows:

1. The **sheet (2)** that contains the worksheet cell that will be set is determined in the following manner:
 - If **rangeName** specifies a sheet (2), that sheet (2) **MUST** be used.
 - Otherwise, the sheet (2) specified in **sheetName** **MUST** be used, as specified in section [2.2.5.4](#).
 - If neither **sheetName** nor **rangeName** specify a sheet (2), the protocol server **MAY** [<24>](#) use any sheet (2) in the workbook.
2. If **sheetName** is empty and **rangeName** specifies a sheet (2), as defined in section [2.2.5.2](#), the protocol server **MUST** use the sheet (2) that is specified by **rangeName** to locate the **range**.
3. **cellValue** **MUST** either be empty and have the `xsi:nil` attribute set to true or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string**, or **unsigned byte**.
 1. If the value is of type **string**, it **MUST NOT** be longer than 32767 characters.
 2. If the protocol server does not support the value, it **MUST** return a **SOAP fault**.
 3. If the protocol server does not support setting a value in the specified location, it **MUST** return a SOAP fault.
4. If there was an application error on the protocol server during the operation, the protocol server **MUST** return a SOAP fault response, which **SHOULD** contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but **MAY** instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FormulaParseFailed
InvalidSheetName
DimensionAndArrayMismatch
SheetRangeMismatch
RangeParseError

3.1.4.30.1 Messages

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

Message	Description
SetCellA1SoapIn	The request WSDL message for the SetCellA1 WSDL operation .
SetCellA1SoapOut	The response WSDL message for the SetCellA1 WSDL operation .

3.1.4.30.1.1 SetCellA1SoapIn

The request **WSDL message** for the **SetCellA1 WSDL operation**.

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1
```

The **SOAP body** contains the **SetCellA1** element.

3.1.4.30.1.2 SetCellA1SoapOut

The response **WSDL message** for the **SetCellA1 WSDL operation**.

The **SOAP body** contains the **SetCellA1Response** element.

3.1.4.30.2 Elements

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

Element	Description
SetCellA1	The input data for the SetCellA1 WSDL operation .
SetCellA1Response	The result data for the SetCellA1 WSDL operation .

3.1.4.30.2.1 SetCellA1

The **SetCellA1** element specifies the input data for the **SetCellA1 WSDL operation**.

```
<xs:element name="SetCellA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** that contains the **worksheet cell** to set. See section [2.2.5.4](#). MUST be present. MUST NOT be empty.

rangeName: The **range** of the worksheet cell to set. See section [2.2.5.2](#). MUST be present.

cellValue: The value to set in the worksheet cell.

3.1.4.30.2.2 SetCellA1Response

The **SetCellA1Response** element specifies the result data for the **SetCellA1 WSDL operation**.

```
<xs:element name="SetCellA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.30.3 Complex Types

None.

3.1.4.30.4 Simple Types

None.

3.1.4.30.5 Attributes

None.

3.1.4.30.6 Groups

None.

3.1.4.30.7 Attribute Groups

None.

3.1.4.31 SetParameters

This operation to sets values of **workbook parameters** in a **workbook**.

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

```
<wsdl:operation name="SetParameters" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">  
  <wsdl:input message="tns:SetParametersSoapIn"/>  
  <wsdl:output message="tns:SetParametersSoapOut"/>  
</wsdl:operation>
```

The protocol client sends a **SetParametersSoapIn** request message, and the protocol server responds with a **SetParametersSoapOut** response message, as follows:

- For each **ParameterInfo** (section 3.1.4.31.3.2) in **parameters**, if the workbook parameter is a **PivotTable page field**:
 - If the number of elements in the **Values** array of **ParameterInfo** is equal to or greater than 1, the protocol server uses the values as the **PivotTable manual filter**.
 - If the number of element in the **Values** array of **ParameterInfo** is 0, the protocol server changes the PivotTable manual filter to the **default filter value**.
- For each **ParameterInfo** in **parameters**, if the workbook parameter is a **slicer** page filter:
 - If the number of elements in the **Values** array of **ParameterInfo** is equal to or greater than 1, the protocol server uses the values as the slicer manual filter.
 - If the number of elements in the **Values** array of **ParameterInfo** is 0, the protocol server changes the slicer manual filter to the default filter value.
- For each **ParameterInfo** in **parameters**, if the workbook parameter is neither a slicer page filter nor a PivotTable page filter:
 - If the number of elements in the **Values** array of **ParameterInfo** is not exactly 1, the protocol server ignores all elements in the array.
 - If the number of elements in the **Values** array of **ParameterInfo** is exactly 1, the protocol server applies that value to the parameter.
- If there was an application error on the protocol server during the operation, the protocol server MUST return a **SOAP fault** response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FormulaEditingNotEnabled
InvalidSheetName
ParameterDoesNotExist
SetParameterFailure

3.1.4.31.1 Messages

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

Message	Description
SetParametersSoapIn	The request WSDL message for the SetParameters WSDL operation .
SetParametersSoapOut	The response WSDL message for the SetParameters WSDL operation .

3.1.4.31.1.1 SetParametersSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetParameters
```

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

3.1.4.31.1.2 SetParametersSoapOut

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

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

3.1.4.31.2 Elements

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

Element	Description
SetParameters	The input data for the SetParameters WSDL operation .
SetParametersResponse	The result data for the SetParameters WSDL operation .

3.1.4.31.2.1 SetParameters

The **SetParameters** element specifies the input data for the **SetParameters WSDL operation**.

```
<xs:element name="SetParameters" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfParameterInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A **session identifier**. MUST be present. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

parameters: An array of **ParameterInfo** elements (section [3.1.4.31.3.2](#)) to be set in the workbook, as specified in section [3.1.4.31.3.1](#). MUST be present.

3.1.4.31.2.2 SetParametersResponse

The **SetParametersResponse** element specifies the result data for the **SetParameters WSDL operation**.

```
<xs:element name="SetParametersResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of **Status** elements as specified in section [2.2.4.2](#).

3.1.4.31.3 Complex Types

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

Complex type	Description
ArrayOfParameterInfo	An array of ParameterInfo elements.
ParameterInfo	Represents a value or a list of values to be set into a workbook parameter in the workbook.

3.1.4.31.3.1 ArrayOfParameterInfo

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

An array of **ParameterInfo** elements defined as follows. The number of element in the **ArrayOfParameterInfo** element MUST be less than or equal to 255.

```
<xs:complexType name="ArrayOfParameterInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ParameterInfo" nillable="true"
      type="tns:ParameterInfo"/>
  </xs:sequence>
</xs:complexType>
```

ParameterInfo: A **workbook parameter** as specified in section [3.1.4.31.3.2](#).

3.1.4.31.3.2 ParameterInfo

Namespace: <http://schemas.microsoft.com/office/excel/server/webservices>

Represents a value or a list of values to be set into a **workbook parameter** in the workbook and is defined as follows.

```

<xs:complexType name="ParameterInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Values" type="tns:ArrayOfString"/>
  </xs:sequence>
</xs:complexType>

```

Name: The name of the workbook parameter. MUST NOT exceed 255 characters.

Values: An array of string values to set in the workbook parameter as specified in section [2.2.4.3](#).

3.1.4.31.4 Simple Types

None.

3.1.4.31.5 Attributes

None.

3.1.4.31.6 Groups

None.

3.1.4.31.7 Attribute Groups

None.

3.1.4.32 SetRange

This operation is used to set values to a **range** of **worksheet cells** on a **workbook**.

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

```

<wsdl:operation name="SetRange" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetRangeSoapIn"/>
  <wsdl:output message="tns:SetRangeSoapOut"/>
</wsdl:operation>

```

1. The **rangeCoordinates** on the **sheet (2)** specified by **sheetName**; specify the location of the worksheet cells to set.
2. If the user does not have the **Open Item permission** on the workbook, **rangeValues** SHOULD [<25>](#) contain only one item.
3. Each element inside the **rangeValues** element MUST be of type **ArrayOfAnyType** and be a **row** of worksheet cells. The number of rows in **rangeValues** MUST be equal to the number of rows specified in the **rangeCoordinates** element.
4. Each element inside each of the **ArrayOfAnyType** type nested inside **rangeValues** MUST be a single worksheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of **columns** specified in the **rangeCoordinates** element.
5. All worksheet cells in the **rangeValues** element MUST either be empty and have the **xsi:nil** attribute set to true or one be of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string**, or **unsigned byte**.
 1. If the value is of type **string**, it MUST NOT be longer than 32,767 characters.
 2. If the protocol server does not support the value, it MUST return a **SOAP fault**.
 3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

6. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section [2.2.9.2](#), but MAY instead contain any subcode specified in section [2.2.9.1](#) or any other subcode.

Subcode
FormulaParseFailed
InvalidSheetName
DimensionAndArrayMismatch
SheetRangeMismatch
RangeParseError

3.1.4.32.1 Messages

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

Message	Description
SetRangeSoapIn	The request WSDL message for the SetRange WSDL operation .
SetRangeSoapOut	The response WSDL message for the SetRange WSDL operation .

3.1.4.32.1.1 SetRangeSoapIn

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

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetRange
```

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

3.1.4.32.1.2 SetRangeSoapOut

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

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

3.1.4.32.2 Elements

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

Element	Description
SetRange	The input data for the SetRange WSDL operation .
SetRangeResponse	The result data for the SetRange WSDL operation .

3.1.4.32.2.1 SetRange

The **SetRange** element specifies the input data for the **SetRange WSDL operation**.

```
<xs:element name="SetRange" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeValues" type="tns:ArrayOfAnyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the **worksheet** that contains the **range** to set. See section [2.2.5.4](#). MUST be present. MUST NOT be empty.

rangeCoordinates: The range to set. See section [2.2.5.2](#). MUST be present. MUST NOT be empty.

rangeValues: The values to set in the range specified by **rangeCoordinates**. MUST be present.

3.1.4.32.2.2 SetRangeResponse

The **SetRangeResponse** element specifies the result data for the **SetRange WSDL operation**.

```
<xs:element name="SetRangeResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.32.3 Complex Types

None.

3.1.4.32.4 Simple Types

None.

3.1.4.32.5 Attributes

None.

3.1.4.32.6 Groups

None.

3.1.4.32.7 Attribute Groups

None.

3.1.4.33 SetRangeA1

This operation is used to set values to a **range** of **worksheet cells** on a **workbook**.

The following is the **WSDL** port type specification of the **SetRangeA1 WSDL operation**.

```
<wsdl:operation name="SetRangeA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetRangeA1SoapIn"/>
  <wsdl:output message="tns:SetRangeA1SoapOut"/>
</wsdl:operation>
```

1. The **sheet (2)** that contains the range that will be set is determined in the following manner:
 - If **rangeName** specifies a sheet (2), that sheet (2) MUST be used.
 - Otherwise, the sheet (2) specified in **sheetName** MUST be used, as specified in section [2.2.5.4](#).
 - If neither **sheetName** nor **rangeName** specify a sheet (2), the protocol server MAY [<26>](#) use any sheet (2) in the workbook.
2. If **sheetName** is empty and **rangeName** specifies a sheet (2), as defined in section [2.2.5.2](#), the sheet (2) that is specified by **rangeName** MUST be used to locate the range.
3. Each element inside the **rangeValues** element MUST be of type **ArrayOfAnyType** and be a **row** of worksheet cells. The number of rows in **rangeValues** MUST be equal to the number of rows specified in the **rangeName** element.
4. Each element inside each of the **ArrayOfAnyType** type nested inside **rangeValues** MUST be a single worksheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of **columns** specified in the **rangeName** element.
5. All worksheet cells in the **rangeValues** element MUST either be empty and have the `xsi:nil` attribute set to true or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string**, or **unsigned byte**.
 - If the value is of type **string**, it MUST NOT be longer than 32767 characters.
 - If the protocol server does not support the value, it MUST return a **SOAP fault**.
 - If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.
6. If the user does not have the **Open Item permission** on the workbook, **rangeValues** SHOULD [<27>](#) contain only one worksheet cell.

3.1.4.33.1 Messages

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

Message	Description
SetRangeA1SoapIn	The request WSDL message for the SetRangeA1 WSDL operation .
SetRangeA1SoapOut	The response WSDL message for the SetRangeA1 WSDL operation .

3.1.4.33.1.1 SetRangeA1SoapIn

The request **WSDL message** for the **SetRangeA1 WSDL operation**.

The **SOAP action** value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1
```

The **SOAP body** contains the **SetRangeA1** element.

3.1.4.33.1.2 SetRangeA1SoapOut

The response **WSDL message** for the **SetRangeA1 WSDL operation**.

The **SOAP body** contains the **SetRangeA1Response** element.

3.1.4.33.2 Elements

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

Element	Description
SetRangeA1	The input data for the SetRangeA1 WSDL operation .
SetRangeA1Response	The result data for the SetRangeA1 WSDL operation .

3.1.4.33.2.1 SetRangeA1

The **SetRangeA1** element specifies the input data for the **SetRangeA1 WSDL operation**.

```
<xs:element name="SetRangeA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeValues" type="tns:ArrayOfAnyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

sessionId: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the **sheet (2)** that contains the **range** to set. See section [2.2.5.4](#). MUST be present.

rangeName: The name of the range to set. See section [2.2.5.2](#). MUST be present.

rangeValues: The values to set in the range specified by **rangeName**. MUST be present.

3.1.4.33.2 SetRangeA1Response

The **SetRangeA1Response** element specifies the result data for the **SetRangeA1 WSDL operation**.

```
<xs:element name="SetRangeA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

status: An array of status information as specified in section [2.2.4.2](#).

3.1.4.33.3 Complex Types

None.

3.1.4.33.4 Simple Types

None.

3.1.4.33.5 Attributes

None.

3.1.4.33.6 Groups

None.

3.1.4.33.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following subsections contain examples of this protocol's usage.

For an example of a system that implements this protocol, see [\[MSDN-WDCAEWS\]](#).

4.1 Getting the Value of a Worksheet Cell

This example shows a simple, and typical, flow of how this protocol is used to enable a protocol client to get calculated results from a workbook. This example assumes a workbook has already been created and is stored in a location that the protocol server can load workbook files from.

In this example, a workbook is loaded on the protocol server, recalculated, a value is fetched, and the workbook is closed. The following UML diagram illustrates this example.

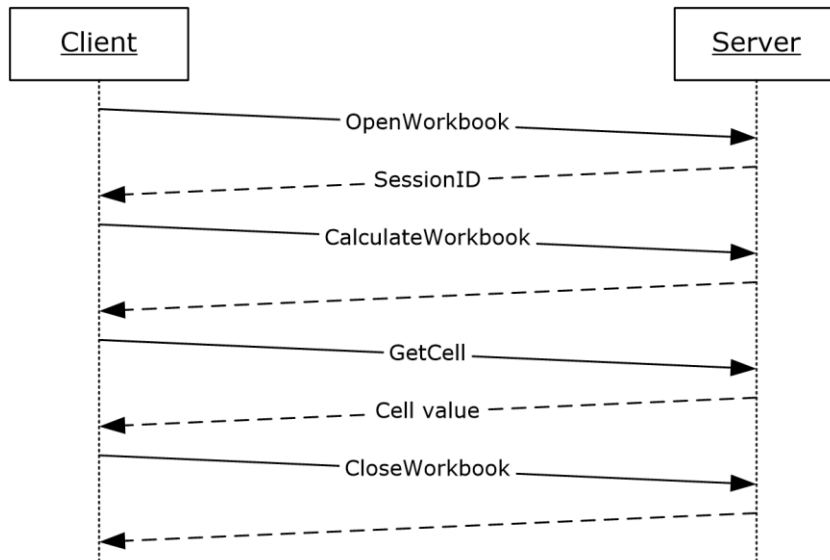


Figure 3: Getting the value of a worksheet cell example

The order of operations, including the relevant **SOAP** requests and responses for each step, is as follows:

1. Load the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <workbookPath>http://server/site/documents/book2.xlsx</workbookPath>
      <uiCultureName />
      <dataCultureName />
    </OpenWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <OpenWorkbookResult>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</OpenWorkbookResult>
    </OpenWorkbookResponse>
  </soap:Body>
</soap:Envelope>
```

2. Recalculate the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <calculateType>CalculateFull</calculateType>
    </CalculateWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

3. Fetch the specific value from a worksheet cell within the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```

    <soap:Body>
      <GetCellA1
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
        <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
          <sheetName>Sheet1</sheetName>
          <rangeName>A1</rangeName>
          <formatted>>false</formatted>
        </GetCellA1>
      </soap:Body>
    </soap:Envelope>

```

Response

```

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetCellA1Response
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <GetCellA1Result xsi:type="xsd:double">5</GetCellA1Result>
    </GetCellA1Response>
  </soap:Body>
</soap:Envelope>

```

4. Close the workbook session on the protocol server.

Request

```

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
    </CloseWorkbook>
  </soap:Body>
</soap:Envelope>

```

Response

```

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>

```

</soap:Envelope>

4.2 Refreshing Data and Retrieving a Workbook

This example illustrates how this protocol is used to enable a more complex series of actions, including refreshing connected data and retrieving the resulting workbook after a calculation. This example assumes a workbook has already been created and is stored in a location that the workbook protocol server can load files from. In this example a workbook is loaded on the protocol server, some values are set into the workbook, the connected data is refreshed, the workbook is recalculated, and finally the desired result of that calculation is retrieved and the workbook is closed. The following UML diagram illustrates this example.

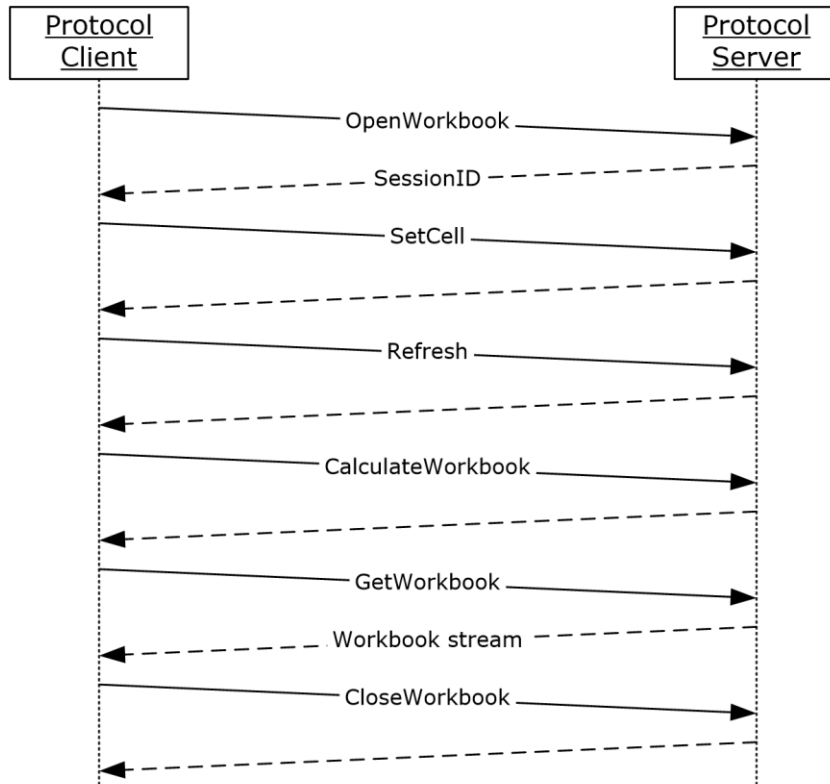


Figure 4: Refresh data and retrieve workbook example

The order of operations, including the relevant SOAP requests and responses for each step, is as follows:

1. Load the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
```

```
<workbookPath>http://server/site/documents/book2.xlsx</workbookPath>
<uiCultureName />
<dataCultureName />
</OpenWorkbook>
</soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <OpenWorkbookResult>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</OpenWorkbookResult>
    </OpenWorkbookResponse>
  </soap:Body>
</soap:Envelope>
```

2. Set the value of the worksheet cell.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetCellA1
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <sheetName>Sheet1</sheetName>
      <rangeName>A1</rangeName>
      <cellValue xsi:type="xsd:int">5</cellValue>
    </SetCellA1>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetCellA1Response
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
```

```
</soap:Envelope>
```

3. Refresh the connected data in the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <Refresh xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <connectionName>Adventure Works</connectionName>
    </Refresh>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RefreshResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

4. Recalculate the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <calculateType>CalculateFull</calculateType>
    </CalculateWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response

BHEDYJo+MpLTI0+T17DIIQs0lNs6RMZxThdaeh+MuzuiScFvuNfGA6woPWNVVPwLwAzMvNeNDC5MbROJs8PL
8RTj2nlwsVN4CfV92g2GbFYqcn+omqBLQ9XpoersUPW6U4E7qcwErAugAnIL/u/DMQjNaM4vmmQ7a4+HJHT
97MA+v3jeDhWLeuWME4R/AvXcmnKpSgTq4Hb6F8SzPchZGHKOn2vSLOOv/0Tg+4Pvj++QT3sP8iqeWvNI91
5HPPCZnCH18oBzUEIPujBU/06ApsTTJyIKQDzgo/fQ9gOpVRAtCXoKiIv9BVGsvoIICSmQHqwtch9wrR0Jn
SJP4ml19MAAnYhXoUHshniI3Ppxmp/UynRJRXYmqIaKdhJZ33Cvo2kPciA3dKmmFTTZKduE73VPH9uq2XcSj
U8TTvOw8D99AePMSWDQsfuSf47N8L14Mxcuh+GYoAhkfdA4JR3tjIcGhJxMaejKBoScTFnryyLlW5F04cg+
mNvBnMvJvMvJvMvJvMvJv0oAVgtfy4m5pR8g43rU/WlCyY4PcwggA5lqOB8cDdvY6hMvpgtMFw2FHRdK5cn
Lk5MjPBvPW+oSgZdasV4aeONBX5obid6uWPZNI83FGl7+dCK+j+lvFnfrxaaWhmPv7KlsGQwMBS4MvbTJm
YEzQ/cMZTAwGBGM49125gzMGZgzMGfA133DbVHODHfZjzjwzHGD2WBR0S02uoAMUECC3l1/+l10fPPbF5r+y
JcbBMV/fL+TgchDwHP/Lvwkxchg5jBx+lPffljLPWXJS4KTASYGTwre+sGca9UzGfOHBiL8zqGXdl5T+590
KtgPHUZ45gMLx+VXj05QutTcGVN3sdnmp/H/h1lrefZnp1+6T8h5L5cPzptgSdqyaEvGmcB0TGGBlrIJCW2
dVieXt80Vy12qsuheVg851j1j9zGhVofozva4I+VpBG25JyErqX42FzftwU0SI58KVbC2aWymLTQZ/dJB/
A8AAAD//wMAUESDBBQABgAIAAAAIQBESXc65wAAAiOBAAAUAAAeGwvc2hhcmVkU3RyaW5ncy54bWx0kmFO
AzEMRO9I/EOU083SAwKUTYVWag9wgvYDzMbTrKqcbewF+veEELaiuPM88j22NVHiuoNC4dMrB5eNFoh9dk
H0R6t11f3WRFAuQhZsJWn5D1y1lEWGZRNuVz6kFkVdG+veT8CKPSJXsc0kgVZAD4EgeB4K0JWUzWlU66k
CQVn2eSFP9p9VE4Thh96ud5eCsuf8rp7gFSNB186aL/eHdBNLTljUd2ZOHYotEAPMQQCe/sxdF6Ae57Mbr
C/Qaw7vKAh69Vg78jn9Q18EBM+O3tQ9Xm2zQPzLmdq1+wQAAP//AwBQSwMEFAAGAAGAAAHAAbclP8r0BgAA
TCMAABMAAAB4bC90aGVtZS90aGVtZTEueG1s7FpBb9s2FL4P2H8gdG9tJ3YSB3WK2LGrbk0bXG6HHmmZ1lh
TokDSSX0b2u0AAcO6YZcBu+0wbCvQArt0vyZbh60D+hF2SEq2GMtI0gzbvSUBEot6fI/ve4/vfaJ87fqjI
FDIiTLccOrXC17iMQ+H9A4ahj3ep0rGx6SCscDzHhMGt6ESO/61ocfXMOBkiQRQTA/lpu44YVKJZulkvRhG
MurPCEX3BtyEWEFLyIoDQq+ArORK62Uy2ulCNFYQZGOQ3d4ZD6BPW0Sm8rU95mcBkrqQd8JrpaNXCfMGNB
qKI15E52mECHMUD8sdPgRz3ySHmIYangRmx+vtHWthZFTSUWtmJub1ze/6bx0wmc0YmKOD81WulU6+s
7XgpJ5J8FkwiL0Ti54vMowYr2KaNqYuDxUORv3gxiLnCfAUKZWlpyzumNqC+45EN1fFSUuMEyQ1srWy/pX4
N35rxZPVPzTrTb7Va7k117n51IV499H3LERjEfjWpno9JcimjkPLaf56PSktfK1SVyxnhgnVmd2yf1zrNzq
y9DiuU8sB+rc85slNeq2yvL44zxwDpTm3Om2txutdaWxxnjgXVmbc6Zznp9rbpEzhgPQkbj0ZwrunF10ssQ
l+n6h5ztFvqyAb5sAM2wxOF9bjAzF4AvTPmHds6Qx2oRG4nwQy46IKAFGTT1GKlJQobYhy7ewlFFUKy5BN4
kOHfHDvlybkbjQtIXNFEN76MEA2ea6Xvz8oc3L5+jNy+fHT9+cfz45+MnT44f/2R1ORN3cRzkJ77+7v0/vv
ke/fn829dPvyyW13n533789NdfvigWBI41W9Grr579/uLzq68/+P7pwXi20Bn8uI9GhGJ7pAjDMAj8M0A
66c9MX5ZvRctJ0ZOATdBarbKnQE70wwK5JrEhe8+wLoZZHqjFFDZ63dUwVLbB8K4wCwT3OWROoXhEAt7St
HMK9cRwUGxfjvNwBxodFt1s4dkLbHifaq7OkDLBvchrZ5j7DscIBiYlC+h4feVLg3QNKHVz3Mn6KHLDUXLQ
Qkh7tO4k0m7RLI4jLpMhnCLWDzd591OSsyOsdguhKwobArGDxPcICGG/gscJRkcoe1jge8NtYhUWL7E6En5
drSwwRDgjjQD0gUhbNuSvA31zQb2EopoVh320TyJUUIo6KdN7GnOclD/foFWJ48ChAoUvjMC97U44gRTHa5
6pIFI+700RfQxxwvDdc9ylxwn16IbhHA2dJswTRd8ZCewGF2qm/EYUHQm7CYswoVGPr/WUxbnjb0JqKtsTu
iRK8SG4JC+80Hsf7BHJ9vvFclt3LuuV95+vuor181mo7K7BQezV5sLzYsORoIUkeUSA6asLlBw14soRmMej
AoJ5njhdJ9FgtCeFjWtwdUUBGMwCJrj6mKuyGOAGOXfG0kKcmqgOJEi7h9M8MF+rW8sDTL1T07rOmnlVSPJF
Z7fGCHV/WwGQdHp2pMywnMCWVmaFUrOKuxlFVUKeh8G2MVvagzW6uYpZlS5libulzoGgx00QQWgoC7Amprc
LCoTcOzCWZkoHG3DTgLi4nCRYZIhnhA0hhpv+djVDFBynLFHBdD7hTESD+EnoJazlpdq30Ha2cJUt5cdYG5
LHrvEqUsg2dRAMontyOL85uTxeio4dVrKzUP+ThpeEN4rIWPUQJRl5r4YRbA2wNfCZv2p25ms8tn0axnjrm
boAKHnhb30YedOpAiqXawDG1qmFtpCrBYW7LrX6kBrBflgm30t1jF6gYk7+2CsDRDS0ZDomv8sHOjWjs7G
VaSvlyEdENB0eoz8biAEP4daqCPwMq4XTCVAR9AS9bNnrml1uc002XP6g3cqDjft4Q01BgloQ47QW6fmrlx
vpiiswUIHOVvw6ALwTWIH9+nE09+j/gnC8Alzhf3GvAk/18Xpw1A4JTONWBVuTDK1yBks7DDY8LFXJoLklI
/y4APmhaAhQBeA8Lt2Gf4tk81+QQ/3f11KrQ2tj8DCvDmiABAwaouJBYD50G1NUTLFWSSmJVZkpMns2tly
Z2GX3ySFhPd3a1jR181AIFcw0ibS6G7mTO9e9TgtjP9DcNV9GndYwZyu2evzThNbWaHDKba+Gp2b4T5dYQJ
bsfDM9o1R5R/SNGXuuZqkExnIdvp4WzLdcwjKz1G1EcX6v1LLFQRTnPYbBKc+ft/EhOn+Al1DhM/utBM2Te
vwAWiaCLxloZZA2kNVXLJ9Euu/ZwT7wYtTokOmrstCmjFijLnGwC36Amdo9AbZe2VnifU6wp5zbnEfsxYsE
O0XYwdqOLYQaIntyi8LQMhs+NYExX2fJf+OE9x9CoHfgzc2YKwMScb5PIjA8UXTNPODNby2aqvt/AwAA//8
DAFBLAWQUAAYACAAAACEAs6QAm6YCAACYBgAAGAAAHhsL3dvcmtzaGvldHMvc2hlZXQXlnhtbIyVS2/iMB
CA75vtf7B8L014NAERqgqElsNKq+0+zsaZEKJnLUNTP9+zzYJMDt+pB8Aez3zssGeYP71VJTMa0kLWKY0GI
SVQc5mJepfS37/Wdwk12rA6Y6WsIaXvoOnt4uux+VGqV10AGIKEWqe0MKaZBYHmBVmRMD2QDNZ7kULXM4Fbt
At0oYJkzqspgGIaPQcVET11hpj7DkHkuOKwk31dQGw9RUDKD8etCNLq1VfwzuIqpl33zwxGVIGIRSmHeHS
Sis82uloqtioX77dozHjLdpsrfCW4klrmZoC4wad6nXMSDAMkLeaZwAxs2YmCPKXP0WwVhTRYzF2B/gg46g
9rYtj2BURgBjK8J0ps/bdSvlrFDYpCRGqnYJGMG3GAJZRlStcTvMJ/zgku0UHQefi4br2t3Y39UCSDnO1L8
1Mev4HYFQbdOnMus9TFb1IJfDMjLBV7c79HkzKc1eLBJA5H0RADb0GbtbC2lPC9NrL6e1KyKXSQ8QmCvy1k
811I4ANySa2YYu5kkeCwld6obZtxNxlgeboXPea4Oie8k8OGD2HD+o3ZlgMtcMsyTn6ZASzEM787Azd9A
VanRnUXfWQ2N9Bw6mrXmQ87c4/2Gj7q0WR6h27v+CrXp2X26aMLutfw9GgS36E/3qRbaZw+vqB7jRM9ic
5V61Umvkm30j79fGe+Ml7jRI+TO3Q7wK4rY6V9+uNF7F7jRJ9G58L1Yp/epFtpnx5f0L2Gp8dJNO20e3QcC
LeCd+I+P+kAvjYn1bb04+r8Nd6DHwK+Xxq2g+9M7UstSQk5PsVwgMVVvuvd2sjGSW1LS4Nt304KHOMaFRUO
sAdyKU27wfa23Bcw+4ZIJXBcuymd0kYqo5gw6GEmpSqtT20c6n7T1n8BwAA//8DAFBLAWQUAAYACAAAACE
AlCPW3KgCAAA2BgAADQAAAHhsL3N0eWxlcy54bWykVN9rZzAQfh/sfxDK61l1bpY0wXZJMgKFRgzawV5lW3
bE9MNIckk69r/vJduOQx/a0ZfkD59d/d9p4tvD1KgZ2Ys1yrB06sJRkzluuCqSvDPP934BiPrqCqo0Iol+
Mgsvk0/f4qtOwr2uGfMIYBQnsF75+oVITbFMOntla6Zgi+1NpI60JqK2NowWlh/SQoSTSzZiilXuEVYyfW9
IJKa3009zrWsqeMZFP9wdAxZGMl/dVUobmgko9TCd0fyEHQ6v4CXPjba6dFcAR3RZ8py9rvKGRASQ01glcie

dRblulAO2ehdqv9wV4JzPMGqb3uoCyhh9GYLGMkQx6a6ncanVJYp3pLF9Qc9UAMTUH+daaIMckAkGwaOoZG
3Elqge7DSiq50LbuyDsC/12c5MBGyNlMIcEzXOJC9G1EOI23pDEQ6phROzigzn461pBegfYtTtH7I7oy9
DiNvg4ukJAWjTntCpilIYGtK40FKx1Ub3i19/901/CbaedAmDQuOK20ogJMcrRRRaoYkI8+nn8VV5gH8qb
MDDZvnuvkTehkc5s8dqDxx+itdgD2BmQ9f+w6FD2+Be3u2l5q6z+PqJ1LY4PjcyY2YWX1c3GBeq7Wj1j1vx
Zu00DVPsV4GX7CFqocC14pSQ7vRIY1/aI9trwF5DBz7mXPazJoexoB6IHal5o2auC/DNI8IPvXsDz7phFWc
OF46rn+awjYBaH82RMfE7nd0SYmT4LsFawkjbcPFUfe3y2v7OCN3LZR/3wtAWIBJ/tez/A07nPWQ7u3sK7h
n/UGJ7gP9Fufb1YLtFj2WS5G8+iRtle76434+3m23Y2ny7m19Hm72BlfWBhbcUKSk5nKytgrZmu2a7F7x7Mv
wYNDW35QBCoGWU5NENsv/PQfAAAA//8DAFBLawQAAYACAAACEAGPZbxGkBAACMAGAAAGAAAHHsL3dvcmt
zaGVldHMvc2h1ZXQZLnhtbIySTU/DMAyG70J8hyh3mo5vTesQEkLsgIT4umep20Zr4sjxGPx7vIwhpF242b
HzxK/fzG4+w6g+gLLH2OhJVWsF0WHRy9/ot9f7k2utmtvY2hEjNPoLsr6ZHx/NNkirPACwEkLMjR6Y09SY7
AYINleYIEqLQwqWJaXe5ERg23IpjOa0ri9NsD7qHWFk/2Fg13kHd+jWASLvIASjZzk/Dz71PS24/+CCpdU6
nTgMSRBLP3r+KlCtgpsu+ohkl6Po/pycW7dn1+QAH7wjzNhxJTizG/RQ87U5NUKaz1ovCrZrVwRdo28n2sx
nZTnvHjb5T6y2u14irraFRdvoettqDnrvy66fSLXQ2fXIz7h5AN8PLMZe/NLVLfU5nmwPj5Z6H7MaoZOeur
rSinb9JWZM5fRCqyYy9hng9gIYlddnWnVIffI++eG+AK+TQvJiUXGm0QmJyXqWF6ZeJNCiLYLN7Z+aafWMAA
P//AwBQSwMEFAAGAAHAAhAJCVG1V8JwAARooBACCaaAB4bcC9waXZvdeNhyY2h1L3Bpdm90Q2FzaGVEZwZp
bml0aW9uMS54bWzcXvtz3Dayfj9V5z+w5mHf1rJkx052LW85kpWk1o691pOtLZ99oGdoiZUZziyH41j760+
DIMBuoAECGGo48YutQV/Q+NBo3MFnf/uyWmaf3pbrqvz2emDh70sqObrRVndnM9+eX/1529n2bbJq0W+XF
fF+eyu2M7+9vX//+fZpvy8bi7y+W1xWXwq7IBBRkoq7bns9um2fz15GQLxFW+fbDeFBVQPq3rVd7Az/rmZ
Lupi3yxvS2KZrU8OXv48MnJKi+rmdTwlzpzEx/rTp3JeXK7nu1VRNVJJSszYcn2ttxsWfL8c3GZN/n5DMpV
F5/qArJcfH93Plv5ZV4ss+ui/gw6gFMTgR2K+ei7756cPnjy8OHjb0/+e7p2el3s+xjPv/tpl7vqsU/dkU
Nok5n2RyK0RSLXWAj5Amnf4lq3K6p3MPv+4LDS1LZz+v8UFqG1aK7e7zWZdn9e7j//RuXRpLxaF82peLC
7rcrku2T9/LhcVcl3e1fMia+42YHnxpSnqKl+CbeuqYAUCyJz8Wgvuky78qi+ViC+Q2y7Nv1J02PavfWfj58
LrItzuA698PPlzstsl6VdRza+S/QXG+EVrPZ5Qyy6rd6mrViNwa7tuyqPN6fgtAnZ2ezbJl8bkAqx+dPX3y
lNqi8lTqcJ4/FOubOt/c3oleYXF9RyyQSe6sod66jAVc29u8LhY/NcVKl/+JSM4+Q5FDs3/wp3yl+euHFwB
LnS/LXNhZPtM/RdmSFF7kVb7otMm/k1Vd1cJRpGHY72RVpXtQaqsW9PNZ9yNZ2S8QJ4pF9neILov1StpH0/
ZVfd1Ae9wSztJJKD5B9Q+/2sbTej1uGaHe2Kr989t6/bnssFbNglJcFPm8s2qgXC4vS5W5evyS7F4D20ba
iS0K2jVktTubYx+2BZsv6JclA1tVfa7xOxH05r9dr1t8iWEpUXr6gp+1BxSiMfTFuJqt1xmp0PQFT6riqAT
QwogAjhy7N5DDuM81NsHwvSmhsFA3dyJAKP/ER69oQNZFauPRQ0NrtXQtlvZCU5Y+Lb1PPhgYIBqkLJmGQX
ApLo60L4XgxHDMekA2pzwADaeOKlLnhDvFoOnI6x43UKhuC8WCxgptZ2OasBdklfnom4or85/nJ/X9bNbs
YGzDh29alm6ThlqWdgyr1a7Zoiuyy3MEWSAy7lASbNBMOmfxWQCG/I1p+yq7LeNtlbmJjc5lviHzYHCY+L6
6sA6eViN2+nqbjd6EQTCkT4OkPM/1lxgTN15hgoUCJXwUSPxtVvoqixE8gUs+wq9aso9I+wlpC9+b2iBe9T
zcJjylcBwOu8LsW8RQz+dmTUQCkMECb1qwDj550Y1Ise4yKvt61fLHCD4OgmMDzPlwbPbbmEYwaVvWgy0SQ
cIBlcHqgszq8CsDfz+W5j9a19gqkIpnwVALy9hXV67BxtglnsLvHrKHHAMhCde6LZasgU9A8wFXu/blfCuj
aNg59STD8wqV+FQ/yrOvlXfyT7GPROEKIhJqG8UiQQiVU226V+sduY6Us9CbGo1OxFNsuAivqXbarSthCk
ittH17M2/0esVrR/YUWqbqkWZY3sKr8Eaa07YmbWb8XDC5FwKD7lu2Xzulu2PgXV7n3rLCBY2081ilJmV17Ei
ixJ29sTOJCs4yxblldrPM767WSxhGn8/E4rjaRttVH/NlulsmVs3V1lcoOnP7bAet6OuJB3FhCoCGaU7HiC
n5HQ4iX0JDIDWRvg9Grd4EhHi5yfAho3vVfSbv9sRbMDK7cbHTwHnkQrS7wU7GgTaCmXm6XoVk4Yld6kYdb
JuFYGLD4qGflhtIR4QLWqFiFdruFCSHCq0atcqdwIm3SaC1JdK/QAgDoLJ0YhaC01ByczA9gazilFoii2JY
elY9C6LlTybUM5HcSe5GYajqd4fS4JLHpaIRMKlFAvOqA2tO4BCoOnSzE4mYsyCvUP1EIEWwyoWrguxqz9v
tqm4UmdWlp+dAbV6X8Ki+vojBiOkNgktr3wsbhqvJzMY7+Ih3/Nnn9pzTr9s85tCDQzFYZ82Waz+OZ/9ue0
LaRqTdcAGkJTrkZ302E76RiSdIFPUuRPd8/rh7agC9D4Abgc60WwIQb2x1hadbHj34xOLrcGRoyjZQyBA0f
2DhIhKvccBhjNnsI4Fbcxw4d6ESMCOuxYfj1MmfPWRXanfeMB+1KcmtbhenHedZ/jFKzcXUHRjAoNFkUmAU
RVxoA3KTgycXGWI3fxIaJZsRlGNNFDD8QBqB9g9r0RlMbiSPNstLs5bo/RMDHS/byKB7VOTgOzF44Dzyk0M
lGs7JqERS1VRrdYlEgvKmoMntPmCfQclJzkPko/zHr9gLFJjznvp6VAMFqUk4UVVxEE2KDsIanRLC6NGKUm
oURVxqA3KxqI28niXbIBh3AghCTaiIQ61IdEpQBOHE6EUF/kSDnDl5ACXID1QhFnWwA5Wu5fmno6bymSAfl
vU5XrBLQMPf8oAtFLWcnhvaOLOmWFM9o9dXsP9mZGkFRWwUuwamRMRtOr4nEzNXZeFozjoPykuMJFoGIbB
KzJoQKyspsoaAcVTIqt31EFdQQM20wim7ZsQbQFFh10UMvIrdTWFzjRCZiHb5V/uBD959YhhxstCvxDDB6
3RhuLmnrJb/myl17Vhpr9E+NdeTB44Q6ALeWZE5B0nB2S4J7YtsAGOWil1bT10flTcaLr6+F4QF/wGrY3rtb3
QOFioiPvhorteY00UmQCfxfBfy54BG1vdNoMzEhiLjciHwZwWJEJSgHDC+c6+wVnpCTScdhofkhsr3tpPF
9aLiOTY4adnaJA13Bwx5RcmbjfgFNqscZsaJ+zjbo8W3pr4vWFOk+g9GRoWsm0B5wm3954mgoJAR0UnwxRN
nZ1brBf7EJKItHa03bJrPEIKmVk0A4JnPNEQk33L6ImFce/UIwv7YuR6IKdEGFiOkJYSyhAXpnd4FmsS3i0
6E5ePnasxGAevfAQ1jdiUUBFLZMMAXqQ6ECJiN/Dag3R35fL249iGQ2jKZHigFgxAt1UnQz6oKIjgwSrG8P
dLwarrvVTZMxqMHwn18OgpiOrCKvX5sFP7cXcNuz3bMdszGdGipI8D2HzE6XRmQXX6bcwyfoFMN5sdWE8KEU
pCEff946wWaYyS02YZ681J6wInNs5mDTkn07BvUGN3Qy5dELQB7X3G+Bypfp4Iw7RvgowBTDQrIigUvQO
m03LqRs7Z41UKYr80fXGLWIS7pkjwJf/+DVg3GazKp70GWWI0DaYGUULS3M6opbBOWPwt4Xqc6tvk6XJjg
UtanLqGxJbJGy86oKLM3+UfwrV6x5VpD7TtQzRFBbw2Qh7A3BcYD39Sciv6gniOC3x9v03cSTOPBnBRvksV
WtRwDnHThzOnDhG0ESI+k+C5ihaOGXToIVSp52D/3SiMxkoc2JJ+YaAU6sLhpNr/ChwdzAWStXNDiU4rJQP8
iVbJTSuwXXsxmqurVzpzDScBVO5duyBD1XMHg7+xK0ApVV+8gNXBiu4VGguXgem0mXm0Lmf7iG4Hb0q5MO
1h+uc4u18tlewzMBQ+j1Fr33KM9Mep5oKxXhDk0WyYTsqlik75CiTMQXe/gCHsmPhuPonqbsXq81yfeVi
A5Z9u7MPlegG3ZLMSnH7sUe9RFsxcAArGmObLcyjplVYbv6mvsmr8r9tu052ea3+wQd6XxEFAkJIQo5oADB
ewKETlFMabkOisajtFTVB2Q3FeAlfObgR0RQaensf23I7niMRQzazSDADDrWxqtaLGBvWlmUkXPETGFxgb1
5qAutSMi2yXagkjVummdhFdfMKo7ZF6GAc1cadUrFwjRMIpTLZ35/gpSFVXET2DQYV00IHSyGQ9syoWbMi

ZloJ/rYryo6PFMR1MFq/jRUxMdnTK8bc4vBQbayXVRztbBeOVdpzjMLw0UptMtnDuk4M2ypWFR+hAfLlZw8
v584UUVW7f98gK3MLqAHFvJ8gNl13DYWLDGuU6n8wWt+Lgt4GR42N3EL0o1JfQ0lmyYhQcgnFgvSj7TmmW
uvenEdIDQ2pqjjoHeXmfK4DohQMbzBSgQU0oSVFQFU3CPQw3KxoI2ctN7CxcLruoCnqcUKyMIN0JIGo1oie
NtSDQWtLHni9aoPHOM7hose3zKJRKLypjzwwt4QrQu4YN1V8v8BvsRIST5EdEQ50dDorGIje1HYB+8TAt3u
LbwRUgGOUpMRY9qiUZwUHxyFMv5b9mrAj5gCc+972oyDr82aGkYgkoiIRyWnhzBBtb/rREYvF3QpaahpsUj
8fLJxSt14iBMPOVAP0IhU1LBEcquihu+dAbZMLChjh7H3ZbMkY/c2IQkXqcouoqc7dIlmJcqvXxA1w5K1JK
EE1UR50eDspMglz/BCaft2a/ihT4EZGms7gSS+Q0hRW2/niuy54R4YHPra6Miedmg1wlojPdcLaePK6HZzk
LlpkGmtw7NXHSib+1NM0E9479fwPaM+j3sKIzrmLwFPP9KEUEHJ4QEHLw8pYnuGz3OGZKfxHtGfKGFJU6JP
Ib22MYWIH441H6q4EGVqmgYOWF5U8NHJGALH76uvGw/e57XMPers/clfPcdPrWswr6A3A8+WcWxhLtiogkS
dk6YW+jj+ALK6c+EryZvVuP1I95soGtrT1y/fcNmVkuDr74s1v1Q4QxOqapY/cdcSy5M9G4Ty+Dr/jgB5Og
cuZuswNqOGek57mBlx4w7jnavSvqtMJM2WsswFY/bLiK1/1FgRz4Ia6w7wsKV7Kkk9V5qqMvY3wNENhhcR0
H6J601vJgAvqpOWVqNBakhvC686iXsmIXrKjDdspBXwSNKFI3XRxO1Dz7gn2QtKxB8OR0zQoHxT4jji8iNJ7
o04mBlavOCXwwZPchyt1/AtZ3HC+AKG3Tewp46bq0LdwxGn6zLjsB5QogMX/VKMhiPwS+pKY2AxzLfkopu
E5KQaDXFwMALRGJw3azbcX/yagfCYr1ck7cWlORCEhatphgseIHpsIBLg9usWWev82r3CT55s6vJiitDTsG
JUQwtVKyDdLXib0hB4hNiWMAWw1FKecUPJSeHsJ4LLXxeJ0xVcbajmZb4coeGpUlo4KtU1FCUBAw4bp
liYPbd9enj0ascjjlnsOPerDEyKDKFHCQeg49XLBKIEVzmFRyNs1tWn5qETK8zIgjledpwrRuIyZptCgTt7D
/ensQeZNN94rndHU8ourjnXiJGNhGrEbv/1e1EsrsSEfwqKk/XSUSj1llgOGInPCE1M1R6oU1+165JMRCIO
rSulDz70fwX3906RycDJurmD3K4xcULUsCbWFRDjxbhPiPR0SP29IJMuZSwkm+7kPpzOwAST4FIghM8TSw
zF/hU6UT6Pui0eqlYusM0zveY8JHh1N/WTJ64FmdK26UfRm1Yx3StT9om/IHM91WFI3IKblJvgqOhfG3ED4
/Wu2Ldbi++XZdwHx1llfISQ0iKJARuobucaEizEamR03X+qWjuslyJnID8XJB3rmlqCmq21hjjoQQnxK/8L
xmUX8PvJIYEnoimJ/Kx+afFIxXVzcgWJilFZNOZyQyLqkJ8Wly8fWiBwXqbElMglMaPSEJazAdgl3TCI5
24KRWI056wXLRjpm3PMMUDqkM8aZekssqUiaXlTrgqLseYjWnkfIu6S/cul1Xe7HBAfSHTB919h5KMLhgyH
cBlql8zUjnc/ejWjMjr25oyeqr0VCWoARmqIQYQUi28sIax/Xu49zZqcQJSfghbTGoOITozw2/yzKmlvSE8
mUFEQ6XRFguCQicRijvazWTbmuvL8v4TMv4py0uE01Fy9ZoZmDSYtCyZ2DRkyyOAJv1LwTuW4LxMIy6Gwsk
O1nbZned0GYqbQ0rJR0H3jCmPLJTYPN6/yLeL+zaowrCTg9DSOsIRanIdmJsCorHiuUnogV0hCN1YdsNFjB
aE23abxmLRNN1OJW93XD7jOjJfVewYkBUyVz51K2LLWmmhB6V7RtxLSEWHfT0Hk1TI3he/h0I+4ie6sFZV/
SWu2pLpe5pCfGzFy01T7jfeBHowpOpP9WtZ5RAUC9XP5S1fAMw8+5+Axyzz0oOQ0qaEKOBg19apoH9fKxrc
4veTiM3hXbYgknjoT9i0Vd0IN9XVI40Iw62awUgrT0LKhqyWRbiIdGi4w00tUqwoap2+TJ7BwvV1Y6EH0pJw
4cqt0tsNjG3YbZsLFryCDc8aZqNT4Rt81qlOGH3qLXp+2AmFdulHkkmJTktXt/n1W/WYRKdmISUlu4Ck6qd
gfbnE4vFqLt1N5JPfb/bio3IrextUQwnhDSsiOootyKZM1DHYjZG67sqazjnJte33+AyaUl4mUqYcntaYoB
0LGrjetqr3AmfQUpCz9ARB96w8LTYvYa5KzyGJZ5mz16sxPoWjv82NQLBW00ciEHyx40jOfXB5sdNf1QnAH
CZStIRRLmfPv90MtPiJ6+hiQ+s4ncC9boFQ07yREZPHJBhCmKxHKMXkTGVfWrPICUhZ+iIQ21YOBYxqXGs8
a/U1n5+BIc/1LwHZLJt1CgFZcwAPS1W8nFANJ5rE5Lwwe8M41DkGYy4RGIXGaPFwfqKuHA40sFOBQBUOFec
5TgDklOgpMuFXEalmV4TsQqtVABc6M8XcKtM/RyYOSm2ERrYGM6kzMnEjA4HIULwHVQJAgRqsXG5QW12i
sC40bqLV1Ui3b2714YrEzs0iDcVjLUSFq9YEGmi09DcmEfpDLfIT2ek8+aY77tfVY00/eH14KMuO6Rumer1
FpwsdbKi11gD9GVBPLGdmbTVEVCxzTULLZBQShhOQZd/Bg5BeMbY17IzThc1fKMjJNOEEQC/u6W8yHHu5xv
N11VCkWHvZjzK2zErPbcU61DMFmwwqFhj32Y1jx3s+Wj31na9yqsgvml+9vS2wpD1ZLXCUKz9LLDSxDSK/H
lqsfoWKy+7krLnftYkPKjhn3djOnW+hub60hyE3aaC3DvDwQXVydb646VbM2v8otSOXuxCTAWAV1HupoS5
jf2zZo46C9E9aSxLoi9u8quC2FdxMgPdFtpn6bbWYjmbWRSRmmZgZihrgvBwPT1hsKx2KOFxpqrG6xSk3n
dFDt9C0BZ2Py1YZfreqPK5YTgkxvSGA0+Pds9B4HT2vLF0LdEE1j3qB+LGrWGJzqYLRJmN5pjQZSMTkhZf
amsZGhgp86IYZZi3FwqJsdserda5fvDn7Q47LvBEsFYC3AI6JYjuteOoWkYR/XGz8FXRXxJxo4bfc7sPabF
5n0UAMouuY3UmzPKopTfrylg8EYbWdvrkrz/GaTpTqP0atjDtwD68ZMq7GuKpgHbVws14A5X+I2mF68Ta
weI94B9eBe0EhQcX0eL4v4EF/8WVUPXLpU1CXIMumScPDIc3K6MWhRfHxH+kws03S5cNW5Srunoy1ZjloK7
w6AAe/6Hu5hpBiiR249+VRE0udolQCdi9g25ZmmIh6gM7jQ/+Her3b2GstPVatw5jIyxzHxH1I4/Gh/q64E
Rd73aEEDhIIdhP36JUVXY92kOiyGLUFdOVyV+7BqWJuS8On09ei+OpPjHqXtg/OSi3KYfAVcmTLA1H2ttzI
jxyKN8iXhXi5hTiYo9Q1FnWwLOd57NTb7+1xABBjdi5DNxFL9cLduIeIuRwBS1q9UEge6g/0WWczx70s13
1Eb7Nws2LhfH58zZPJ/ffj/C93ryen57B3R07exa6+7LIq701g08ogynP8UGrR8yk9v0m0RATbvJAA8qOR
LAR4tVsQ1H3GfC3JTbXrmglqOBPQA1xYsY4HbZid7gbogIdugKTKtiHDQ11zn02HkgaA4IYvvpq1FganMza
bBW5YyEohHDQXVvD4W9a/5cleIQ84N3Jk+nz3dI9jCM/rWjTWMoqQ6kFqs0BMcukziwHMI7QVgcq8kgIKO
yDoIR8FShcMAplRFRT9ipYn2pMD9syjI+VULN8EwDmxtVmbh/e20rVtWb1LQ3H1DV9Ej9AxIU4qrsf3JNKB
dwvNMOX2JVAWH5IWPa7uVMY4kOMkGhJ18QgjaqnKGoXi2ZYB1Mm7Z8N0GBAPq2mX5bFHALM18PthbDKPA7
Sp1YLJFw27yhrUcQRQu+NjV4gR4iPS1ALjOPFRGoFHgskDmXYE459TE5Y9PZLoigLQLxntfaOAJ4YI fuwx
57QYVVRyHkFDwjcegdPImbqA4xQhGsJBudGSjKh80w0qCCTi+xmaN6OjYQ9dDlwnTRaS2fpbmlottCFTY8F
nR1TiZzu4hwzNabqOCLASoip6Tw0FXQLQio1U81OMEL1YJFL+sqBhP0Znix15XV0BAqVj6HrpW2XfAV1+2
62W25BSVMAEwPg17V0jDjeGimMEEJWqweDgtNKEf2IYU7FKt3BuYUjJ6QBSPaHoDUjFQCDBjeh/Nw3aIV
DHJOq+mz5mir43dePRJHZOsUnBkwM18YCG5XE9KR2YxkcoWh6JGKKBG8a9iWX4u4AyqGUK4Nop5R+k0sEIAT
wVu1Q8N4xmrwAEvUXO//QnJChVZNX3ctDFzD7hiTXFZrBejrnSmQ+7UcHSYG4HeCTjLgWftqjAa6gHxJJyl
zvuiJ212i2GYFAM4+GsNKYC7ZQ/MqRRd+IEuecZAd9eWTS0HtEkVMfx3A0c5BBvyUFxLgv9g4CpUk34Iia
kSDXJrW7Seyp3LqanUhs5WReQgggWcgo+Ohq+U1IPoIF91+3dP32ALPs7/bwmQzZrds+ndGQJ8VBMHJKK5r
DTVsDDqdVYYL6pb/Kq/G8uPsAxhvn+UKxv6nxzK04U679NTANBRDPCR7UeKzMJjKLz61OKapvKsfPA9josZ
Mda86cKdHbVReO4GM2R8ayBo+8BN6cOXdpMyFax4twxR+LcnePbLwe+2rz53EtN4nYGoJcFKHTj9D1qCqsx
c5M1gjm4msD0MA080kTPvSsrXnIAK3VrJao47vDtVC0xvGQOP0WzrLJFeeC0gnsFzRFYVOaYjAMLR9/DXTL

1qNlZMBVBBA05KsOXKD9cvDXGpvPvdSkHJ+4rnpKansBCtcfSt6j2sKzlpWFctXYQ/ZRN3GRjs4FUARsYl
ePz+L7qR11FASmveH0q+sFIxT3+wNcJpdZP83w9oZWxmUqqyLytKpaXzOFHSjrQluYqtKNMThii6mnwX5wF
dtY6Cp9nU9GwuuUPhp8jdUcFlzKsyeyVfKUrAOi0ZhKfWN7bGelrnofpIppHEyVthRQnbJHgiPasWEB7e17
YtkrioLRiXaN4Bge+rpe6tnVhbJWbMyJ7A8VsAyzbCVVYptV5guAHbG9uXg+oxqQ/QP72pyV8rUI6Jisjzr
qXCjLZJa+/NLALbVlwXz7QZtq8Exm63v4UKan5nvYzBbCc8TiQ7lweoV+xF1DiRkms/L9uoFvZKmXwp222m
yTWQzFhNXKRfDRrOck9ktG/mF+VUV7Q6InmZjqwBej9BHx8OD1X5KzKPUtIMUzqwbdm6Kq7KYgnXRE9nz
599En9uf9nmN4WYXyfl1C/dfdITxPr82Qm9Xuq/boqiszEdtyMjYxVfKohqVq0CtpB4Amyg2UyS43+gTXV
4JnM1r6TYM3syZNziLsI1kbMcGgrL9U3003WY5A0bZluFHa3xaJos01mMdtNsUaznJPzjXov1lpET7Nxj97
NHT0pJcowskKXFN2NMihm0gHNoz1LGweg5PcI+02iy20qQc2T7qHHRlx+oFN6vszgpSeSB3YhNc1YYNw+o
FNQkEKW4SSEwXqpfHQN3xYjP0FFuNhgobNsQkjl1r3P65uiSTPP9m6ZEmXGVVnBczFl+6HmzbpuyuomKZq+g
PO2YvbwzngcAqdHGfbyyx3e40pgH0slqIa2PirrVplkE6tGV61gkLXvBVXpj/fyrJxz5ZhigrtYbWT5Pq
VKvAVmavwVWL81axj28ym5Xj/VKVDQBczsk5eW2yzTa5xTiw4IBj2YwZJ7MaWla5zpp1JrbZu4UssbOuv9+
hzWY5p7f7bV7DIUllut9yynto26EZ/7tcZvCVYLWkAx+G2xK/drAc2lI9A8SRA0PLMKRzqTWkxzetaLvJxD
cf32Q224GLxdddmm9xiHLA8NmPGyaxmoxZrNs5vd00ZvktP7yHtl0vW8GxCHhFsoHxJDaXIR/aQrs1YQnta
Pr90t27E1pJAZZLAG44nIGYfmATcajFpuH0CUIo+j2RCMf2WUC2MXX+CjJdtCDLLEFz+Latd0s+ARVzTKvEc
2FQ2BmI7WYapjKThjJWtshzYUD10HpjE90QoA7X6Nh6giBN+Pwb6W1t7+u82uZzcVMEdt/hZV0MrJtrQtN
1PJRfZbJNTqlRpmJxvI6RBLIeivZVjtFlyFG2av1RPKK3MTP7bmy0h21C4zUUXnd2c01oeg8oD3OUadJ97V
FJkvt6sbSIB7ZTvkWwgNcmSLzSibNsW8Ab2/A4+KZdMAC+89m331kn13EYpc+Foce5Yyx2wS80Plz8btam
Mhew8NCy2129a/s7OHDm1yzbi7W60++tYyGwpDQdH+GPwoy/NHxGf44yPDHkxn+a13dZK+KXB/Mi0twNnUy
Qy/WdaGdum0IGFmbyhp6+hAeul/QU6Sj0/ErsTmjrw5RPMJt/GRHR1Gt/FHOOSIetd6vVqLGRcx1sXDW31
2AGTVNaS3f8pM7++wg5gkFgbITzcd92jge3PcK/2p2q+XhVz+wY4tran0c5qJQ87d3F+rE22PrfshzV9dN
cgHdga0JHQ7IjP1IXskkVpUztdV240eQod672C9hxX+BV9TWUcTNRQdvDY0/Ox+bHuzgdLatd0s+ARVzTKvEc
XJoS1oLD2SV6WSmUa0XTmKZ7XCWbV63g9f7D4Sh5XuWoRN7YldytPX5oOLRqB3EC1WuVrDrISQsElnn80Vt
39TeqA3x+CP1oWjmqAgj3PtpOX0nKVdc5TKd2yt5Pm3VPXomn7Plnr62Ca20vdRh5zSe6jDG81YfH0VXe+y
99eLvImZxV3CVInux3cIM2vZV4Naed/tyMzT8k+WYRLlBj+0bPN64yLMVQ4EouW1lqW+bvyb+7J73RyTib
MbTHaR0xy9DfFqG5kQ61oFDD9M6cvd9oNhbmqtbuy2S9yv6Zzf+Cz3uYgVpfa3u4+Vv8I3H8VXAe0IZfnQY
O+nedm5W1HKWTKRdbaHMvb5Y5W2fORFACyQyldPNqmtsnrRnRvY7xuhGntZFu172WkV1MP3b/VOLO2fWCA
ldqp0YVVwYPzBhmC02RfbLL8YX+U1UEG53P6Rqne+mQVE52tVIKtULX4n10NCpj9Y1wWGQnCbYv5Iu4AR
+VYjXs+SpExsqnsNhoQ2Z3j5tbUVbZ+HjCtMCCzuWwWGHheEIBuoTDwpF0+VZBmogjRdaIB01rUIZZZVetg4
Ua5mkOwN5EK63K1ZmocvjCmFW7WjrdROjMuTPZVofm460Qj17XrqztKvdzUjP7mrdW80RvssBXCp5gf0i
QN4Ifb5s9825VY/k3AGzydAAvmydr/5gU7o4URj96ofnsHJ1+2mgLNTn4v/e52XcAYWTVu/EXer4HaFnNdt
dT1A2cUpl76XaGvcm1DCuezWfZZfL2fPbhd+ET2+KnhDcwH6S6k6mLWNT218fJp2f2EdstTjZVhb+3770
AhRDrTQD7BV6Ytsh+CILRjH00zB43FlbTNA+4FqsHYovXB7TFHAD3NzweH9P3d3qNPFfwkPe5K09g/LlnUi
YGQa+Zg8DX3GHwa/aQCuibMLRpuBAoWzqqBBfDkVUEb6anMhwCngpxSPgqxSESUjHmKh+qEpsUVBmiw1ID6
v0juWmEB2qL1QOyxuE12IOAdazTIYw9n1FWR3aibYjwEwFc/JawgLGryv/sCnF/B5aifKw4PxfNaFbpm8e
fHhVdI3E7VciGqCiwZVwLj+bpvh8XiG2ePzDLCpbYY9B0SuW7BWwBDUhbYDcDnOFu/hoVEjnFSqYARXZF2
nuK6A4eWnT+W8LKr5nRlyYvzeZRicrLgD38Xs83ynjK8inEThldfNGxH6OsUUA69vjLbPu2Oglbz6RQcbTj
W/9TiwYvGBpXhCsDEWThBGfMvvrMx1Fm/8pcz905Q2ahhW/YLHl0m5RokyH4YmbwiWekqs3UuPr23S3mhqL
W351Y11x2SQMvVQ1JzKizZ0aPH6ELWYQyAVTY+5DY/clM1hAKxv0+OWDWEFhrqkN+3N4KntDshZyQ0bPB28
vtjpeVhH7pLh4D8Rqx7Pn+nXh/UKyJlYAdhJWQW3KM5nL+btM5gzsiZyoUvFJzEVIYwCmDpUtRhKVDLWotL
EwKp5hJepDFUilCqs3rJwovjDgqYOTCPKMIe1SH+My1IJIYU9+sgt5cf+RLF7ZFs6Rtp910t/8ttTaLNRgm8
5lpT40ZahXyVilSuPU6I+vGhp00lakEz1ntP118p2oYwLWzJawgLGryv/sCnF/B5aifKw4PxfNaFbpm8e
qTPqWU2I2uVjLNRaVSlVIO/hGOowiSsdqdzKruZgaGtS8WKFCTPFoAkrxbYwms6oadjZjSp7ttQpJKxHpXG
FUrRuE+oOFRzrF2HB9ngHxedOMBI09Cw5kQglurvi3KKpUkW2cnwkAu81RXK3wNxcNp28grdOavv53H1kp
T7Xw0iYULnoqA4ylrU2mXTLQpVs5E9TESU49KJ4pUImvQegfPV7ii8TWLEq2UJHSf6N4M+4t4u5ECvvdvo7h
cuUFBK1wNb76flG3Ex6HxmECATrLeT7mc7ae9BD1k5Kq9Hx1PVQ+OVZYY4oEU/5qcKo1V01AH51rtR36rFJ
YGX1oHGyt2kDhibuWtxX+4tTfzGDuSzoCrWWRQsEVVuk9WL4fpmpc6VbpPtn/ZUNWV1FXpIHuCHU/4d967
9bdPhFvng+0oThs3DT5qum1d+Uw0kSGmb3gm2LXUbaU7iWtropy07CjT45DsgmyC4xjs8Wh+P9/xXGekfKc
hTI7yUU2PQJR9G8IEkYstH83vNmIq0ydBuoLsgguQAXadOax/FI17ZXKUKDI5cKdMjvJRJhfuBleqQVS7cdV
1Bdrlwp7pcuMMZNdSeHZVDmRwlpEwO3CmToz1TJgcIlm1VOQZxkOmuyjF0Bdn1qhyqy1U5cPIGVY7DeMrkw
J0yOvo9ZXRKN2fk5MLd4Aoy/TTI91NHT2Hk6KgyduWqHYPL0SgehNQOZXKUKDI5WgV1clQhZXJUIWVYVaHB
5ahCgyuogK4qNHQ5qpByuarQ4HJUIdxnHW5g1M1RQsrkwtTgcmBqcDn8weBy1LXB5cUCrkwNbgcmLbPpOi
xpiNDyuSkU01DEFqVay3K5KgeyuRALDK56tDgCrLqNCxHRx0a0TogpVyuOjS4HHUoJhwpNwyDyGSDK7T1v
wTMWmBqcyM/LxuLvL5bXFzfCrhzWmYoDz/fwEAAAD//wMAUESDBBQABGAIAAAAIQAIKrfqzAEAAngCAAASA
AAAAGwvY29ubmYjdG1vbnMueG1sbJNbxpBD1bvlfofRlMxXRK1agNdHkCRkLICdUnTA5dhaXrT5mMaXh
38dL2yRFva09tt/Xj3d8++QdHJGSjUGK4aAuGEFHY0MjxcP67sNnASmrYJSLAaU4YRK3k/fvxjgGgDpzWwK
eEZIU+5zbm6JIEo9epUFsMfDLLpJXmUNqitQSKpP2iNm74qosPxVe2Samb8aBNWxeQDT6zjqWnN1sHhI73P
yM+6CVbzfzqDuPIadNdYK5ygrq2JHGtHly+JSNyl9gao5cORHCY6RDgVnjZWrACGI0iO3U2SMLsWhQnj8ue
gXkU8vpjwIId4Ts3z/i+xawFbpQ00xc7+NJ3XE3tR5oCY07MMqx4BSjhwyUKbrsLZNWIYFNy2/YRspJ+Zg
tiuCV7ZSrCgerUGSVb28n64G16NFyNiQymigRt2RzSdZ16vFaNV7svklC4uwi3JNHXKP7T3Ajh252MiLDZn
O6A1H+YJxVM1/wCz6VmW7ta6XGo5qtCn8gm7vr68OhdVNvRQYV+iwRVNCi/EkUsvI73/Af9h+uf1/UZ71
AUk3F0qmUCCY05j1r19+ezUPw1J+vcjCFnzPrL2RcXr6D+CdLkGQAA//8DAFBLAWQUAAYACAAAACEALmnZ2

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#+006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
    </CloseWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

There are security risks associated with exposing **session identifiers**. If the session identifier is exposed, it is possible for an attacker to read information from, or modify data in, a **session (2)** on the protocol server. An implementer of this protocol is to consider keeping session identifiers protected. There could be some cases where it is desirable to expose the session identifier, but implementers are to use caution in how they expose session identifiers and consider the security risks.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices">
      <xs:element name="GetApiVersion">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetApiVersionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetApiVersionResult"
type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="ArrayOfStatus">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Status" nillable="true"
type="tns:Status"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="Status">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="Severity" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="NewWorkbook">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="path" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NewWorkbookResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="NewWorkbookResult"
type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OpenWorkbookForEditing">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```

```

<xs:element name="OpenWorkbookForEditingResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookForEditingResult"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="OpenWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="OpenWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookResult"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CloseWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CloseWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Refresh">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RefreshResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SaveWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SaveWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

</xs:element>
<xs:element name="SaveWorkbookCopy">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType"
type="tns:WorkbookType"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions"
type="tns:SaveOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="WorkbookType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FullWorkbook"/>
    <xs:enumeration value="FullSnapshot"/>
    <xs:enumeration value="PublishedItemsSnapshot"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SaveOptions">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="AllowOverwrite"/>
        <xs:enumeration value="LockAndCompareVersionsOnSave"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
<xs:element name="SaveWorkbookCopyResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SaveWorkbookCopyDetached">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions"
type="tns:SaveOptions"/>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SaveWorkbookCopyDetachedResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Calculate">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="RangeCoordinates">
  <xs:sequence>

```

```

    <xs:element minOccurs="1" maxOccurs="1" name="Row" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Column" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CalculateResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CalculateA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CalculateA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetRange">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetRangeResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRangeResult"
type="tns:ArrayOfAnyType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfAnyType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="anyType" nillable="true"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="TypeExporter">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="CellError" type="tns:CellError"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="CellError">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Div0"/>
    <xs:enumeration value="NA"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Null"/>
    <xs:enumeration value="Num"/>
    <xs:enumeration value="Ref"/>
    <xs:enumeration value="Value"/>
  </xs:restriction>

```

```

</xs:simpleType>
<xs:element name="GetRangeA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetRangeA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRangeA1Result"
type="tns:ArrayOfAnyType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCell">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCellResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCellResult"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCellA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetCellA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCellA1Result"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetPublishedItemNames">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="WorkbookItem">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ItemType"/>
  </xs:sequence>

```



```

</xs:complexType>
<xs:simpleType name="ItemType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Chart"/>
    <xs:enumeration value="PivotTable"/>
    <xs:enumeration value="Table"/>
    <xs:enumeration value="NamedRange"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ArrayOfWorkbookItem">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="WorkbookItem" nillable="true"
type="tns:WorkbookItem"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetPublishedItemNamesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetPublishedItemNamesResult"
type="tns:ArrayOfWorkbookItem"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSheetNames">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="SheetInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:SheetType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Visibility"
type="tns:SheetVisibility"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="SheetType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Worksheet"/>
    <xs:enumeration value="Chartsheet"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SheetVisibility">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Visible"/>
    <xs:enumeration value="Hidden"/>
    <xs:enumeration value="VeryHidden"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ArrayOfSheetInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="SheetInfo" nillable="true"
type="tns:SheetInfo"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="GetSheetNamesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetSheetNamesResult"
type="tns:ArrayOfSheetInfo"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetRange">
  <xs:complexType>

```

```

        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates"/>
          <xs:element minOccurs="0" maxOccurs="1" name="rangeValues"
type="tns:ArrayOfAnyType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetRangeResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetRangeA1">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="rangeValues"
type="tns:ArrayOfAnyType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetRangeA1Response">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCell">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
          <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
          <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
          <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCellResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCellA1">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SetCellA1Response">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

```

```

</xs:element>
<xs:element name="CalculateWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="calculateType"
type="tns:CalculateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="CalculateType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Recalculate"/>
    <xs:enumeration value="CalculateFull"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="CalculateWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSessionInformation">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetSessionInformationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="serverVersion" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureNameUsed"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureNameUsed"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CancelRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CancelRequestResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType"
type="tns:WorkbookType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="GetWorkbookResponse">
  <xs:complexType>
    <xs:sequence>

```

```

        <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookResult"
type="xs:base64Binary"/>
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SetParameters">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfParameterInfo"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfParameterInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="ParameterInfo"
nillable="true" type="tns:ParameterInfo"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ParameterInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Values" type="tns:ArrayOfString"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfString">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="SetParametersResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetCalculationOptions">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="calcType"
type="tns:WorkbookCalculation"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:simpleType name="WorkbookCalculation">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Manual"/>
        <xs:enumeration value="Automatic"/>
        <xs:enumeration value="Semiautomatic"/>
    </xs:restriction>
</xs:simpleType>
<xs:element name="SetCalculationOptionsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetChartImageUrl">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="size" type="tns:Size"/>
            <xs:element minOccurs="0" maxOccurs="1" name="chartName" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="Size">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="GetChartImageUrlResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetChartImageUrlResult"
type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="EnsureWorkbookModel">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="WorkbookModelInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Server" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Database" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="SessionInteractivityTimeout"
type="xs:int"/>
        <xs:element minOccurs="1" maxOccurs="1" name="Version" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="EnsureWorkbookModelResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="EnsureWorkbookModelResult"
type="tns:WorkbookModelInfo"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="OpenWorkbookEx">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
            <xs:element minOccurs="1" maxOccurs="1" name="exclusive" type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="OpenWorkbookExResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookExResult"
type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetWorkbookConnections">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

<xs:element name="GetWorkbookConnectionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookConnectionsResult"
type="tns:ArrayOfString"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RefreshEx">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="targetApplicationId"
type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RefreshExResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="RefreshDetached">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionDetails"
type="tns:ArrayOfConnectionDetails"/>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="ArrayOfConnectionDetails">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ConnectionDetails"
nillable="true" type="tns:ConnectionDetails"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ConnectionDetails">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConnectionName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TargetApplicationId"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DetailedRefreshResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConnectionName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="ErrorMessage" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfDetailedRefreshResult">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="DetailedRefreshResult"
nillable="true" type="tns:DetailedRefreshResult"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="RefreshDetachedResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
      <xs:element minOccurs="0" maxOccurs="1" name="detailedRefreshResult"
type="tns:ArrayOfDetailedRefreshResult"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    </xs:complexType>
  </xs:element>
  <xs:simpleType name="RangeReference">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="SheetName">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
</wsdl:types>
<wsdl:portType name="ExcelServiceSoap">
  <wsdl:operation name="GetApiVersion">
    <wsdl:input message="tns:GetApiVersionSoapIn"/>
    <wsdl:output message="tns:GetApiVersionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="NewWorkbook">
    <wsdl:input message="tns:NewWorkbookSoapIn"/>
    <wsdl:output message="tns:NewWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="OpenWorkbookForEditing">
    <wsdl:input message="tns:OpenWorkbookForEditingSoapIn"/>
    <wsdl:output message="tns:OpenWorkbookForEditingSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="OpenWorkbook">
    <wsdl:input message="tns:OpenWorkbookSoapIn"/>
    <wsdl:output message="tns:OpenWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CloseWorkbook">
    <wsdl:input message="tns:CloseWorkbookSoapIn"/>
    <wsdl:output message="tns:CloseWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="Refresh">
    <wsdl:input message="tns:RefreshSoapIn"/>
    <wsdl:output message="tns:RefreshSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbook">
    <wsdl:input message="tns:SaveWorkbookSoapIn"/>
    <wsdl:output message="tns:SaveWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbookCopy">
    <wsdl:input message="tns:SaveWorkbookCopySoapIn"/>
    <wsdl:output message="tns:SaveWorkbookCopySoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbookCopyDetached">
    <wsdl:input message="tns:SaveWorkbookCopyDetachedSoapIn"/>
    <wsdl:output message="tns:SaveWorkbookCopyDetachedSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="Calculate">
    <wsdl:input message="tns:CalculateSoapIn"/>
    <wsdl:output message="tns:CalculateSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CalculateA1">
    <wsdl:input message="tns:CalculateA1SoapIn"/>
    <wsdl:output message="tns:CalculateA1SoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetRange">
    <wsdl:input message="tns:GetRangeSoapIn"/>
    <wsdl:output message="tns:GetRangeSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetRangeA1">
    <wsdl:input message="tns:GetRangeA1SoapIn"/>
    <wsdl:output message="tns:GetRangeA1SoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCell">
    <wsdl:input message="tns:GetCellSoapIn"/>
    <wsdl:output message="tns:GetCellSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCellA1">
    <wsdl:input message="tns:GetCellA1SoapIn"/>
  </wsdl:operation>

```

```

    <wsdl:output message="tns:GetCellA1SoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetPublishedItemNames">
  <wsdl:input message="tns:GetPublishedItemNamesSoapIn"/>
  <wsdl:output message="tns:GetPublishedItemNamesSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetSheetNames">
  <wsdl:input message="tns:GetSheetNamesSoapIn"/>
  <wsdl:output message="tns:GetSheetNamesSoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetRange">
  <wsdl:input message="tns:SetRangeSoapIn"/>
  <wsdl:output message="tns:SetRangeSoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetRangeA1">
  <wsdl:input message="tns:SetRangeA1SoapIn"/>
  <wsdl:output message="tns:SetRangeA1SoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetCell">
  <wsdl:input message="tns:SetCellSoapIn"/>
  <wsdl:output message="tns:SetCellSoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetCellA1">
  <wsdl:input message="tns:SetCellA1SoapIn"/>
  <wsdl:output message="tns:SetCellA1SoapOut"/>
</wsdl:operation>
<wsdl:operation name="CalculateWorkbook">
  <wsdl:input message="tns:CalculateWorkbookSoapIn"/>
  <wsdl:output message="tns:CalculateWorkbookSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetSessionInformation">
  <wsdl:input message="tns:GetSessionInformationSoapIn"/>
  <wsdl:output message="tns:GetSessionInformationSoapOut"/>
</wsdl:operation>
<wsdl:operation name="CancelRequest">
  <wsdl:input message="tns:CancelRequestSoapIn"/>
  <wsdl:output message="tns:CancelRequestSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetWorkbook">
  <wsdl:input message="tns:GetWorkbookSoapIn"/>
  <wsdl:output message="tns:GetWorkbookSoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetParameters">
  <wsdl:input message="tns:SetParametersSoapIn"/>
  <wsdl:output message="tns:SetParametersSoapOut"/>
</wsdl:operation>
<wsdl:operation name="SetCalculationOptions">
  <wsdl:input message="tns:SetCalculationOptionsSoapIn"/>
  <wsdl:output message="tns:SetCalculationOptionsSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetChartImageUrl">
  <wsdl:input message="tns:GetChartImageUrlSoapIn"/>
  <wsdl:output message="tns:GetChartImageUrlSoapOut"/>
</wsdl:operation>
<wsdl:operation name="EnsureWorkbookModel">
  <wsdl:input message="tns:EnsureWorkbookModelSoapIn"/>
  <wsdl:output message="tns:EnsureWorkbookModelSoapOut"/>
</wsdl:operation>
<wsdl:operation name="OpenWorkbookEx">
  <wsdl:input message="tns:OpenWorkbookExSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookExSoapOut"/>
</wsdl:operation>
<wsdl:operation name="GetWorkbookConnections">
  <wsdl:input message="tns:GetWorkbookConnectionsSoapIn"/>
  <wsdl:output message="tns:GetWorkbookConnectionsSoapOut"/>
</wsdl:operation>
<wsdl:operation name="RefreshEx">
  <wsdl:input message="tns:RefreshExSoapIn"/>
  <wsdl:output message="tns:RefreshExSoapOut"/>

```



```

    </wsdl:operation>
    <wsdl:operation name="RefreshDetached">
      <wsdl:input message="tns:RefreshDetachedSoapIn"/>
      <wsdl:output message="tns:RefreshDetachedSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExcelServiceSoap" type="tns:ExcelServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetApiVersion">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetApiVersion"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="NewWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/NewWorkbook"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbookForEditing">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookForEditi
ng" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CloseWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CloseWorkbook"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Refresh">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Refresh"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>

```

```

        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbook">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbook"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbookCopy">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopy"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbookCopyDetached">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopyDeta
ched" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Calculate">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Calculate"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CalculateA1">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateA1"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRange">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRange"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRangeA1">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRangeA1"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCell">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCell"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCellA1">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCellA1"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetPublishedItemNames">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetPublishedItemName
s" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSheetNames">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetRange">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRange"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetRangeA1">

```

```

    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCell">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCell"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCellA1">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CalculateWorkbook">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateWorkbook"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSessionInformation">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformation"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CancelRequest">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CancelRequest"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetWorkbook">
    <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook"
style="document"/>

```

```

    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetParameters">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetParameters"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCalculationOptions">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCalculationOption
s" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetChartImageUrl">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="EnsureWorkbookModel">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/EnsureWorkbookModel"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OpenWorkbookEx">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookEx"
style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetWorkbookConnections">
  <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbookConnectio
ns" style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>

```

```

        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RefreshEx">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshEx"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="RefreshDetached">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshDetached"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ExcelServiceSoap12" type="tns:ExcelServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetApiVersion">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetApiVersion"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="NewWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/NewWorkbook"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbookForEditing">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookForEditi
ng" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
    </wsdl:operation>

```

```

        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CloseWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CloseWorkbook"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Refresh">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Refresh"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbook"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbookCopy">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopy"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SaveWorkbookCopyDetached">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopyDeta
ched" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="Calculate">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Calculate"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:output>

```

```

    </wsdl:operation>
    <wsdl:operation name="CalculateA1">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateA1"
style="document"/>
        <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetRange">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRange"
style="document"/>
          <wsdl:input>
            <soap12:body use="literal"/>
          </wsdl:input>
          <wsdl:output>
            <soap12:body use="literal"/>
          </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetRangeA1">
          <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRangeA1"
style="document"/>
            <wsdl:input>
              <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
              <soap12:body use="literal"/>
            </wsdl:output>
          </wsdl:operation>
          <wsdl:operation name="GetCell">
            <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCell"
style="document"/>
              <wsdl:input>
                <soap12:body use="literal"/>
              </wsdl:input>
              <wsdl:output>
                <soap12:body use="literal"/>
              </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="GetCellA1">
              <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCellA1"
style="document"/>
                <wsdl:input>
                  <soap12:body use="literal"/>
                </wsdl:input>
                <wsdl:output>
                  <soap12:body use="literal"/>
                </wsdl:output>
              </wsdl:operation>
              <wsdl:operation name="GetPublishedItemNames">
                <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetPublishedItemName
s" style="document"/>
                  <wsdl:input>
                    <soap12:body use="literal"/>
                  </wsdl:input>
                  <wsdl:output>
                    <soap12:body use="literal"/>
                  </wsdl:output>
                </wsdl:operation>
                <wsdl:operation name="GetSheetNames">

```



```

    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetRange">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRange"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetRangeA1">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCell">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCell"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCellA1">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CalculateWorkbook">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateWorkbook"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSessionInformation">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformatio
n" style="document"/>

```

```

    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CancelRequest">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CancelRequest"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetWorkbook">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetParameters">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetParameters"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetCalculationOptions">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCalculationOption
s" style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetChartImageUrl">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="EnsureWorkbookModel">
  <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/EnsureWorkbookModel"
style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>

```

```

        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    <wsdl:operation name="OpenWorkbookEx">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookEx"
style="document"/>
        <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    <wsdl:operation name="GetWorkbookConnections">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbookConnectio
ns" style="document"/>
        <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    <wsdl:operation name="RefreshEx">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshEx"
style="document"/>
        <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    <wsdl:operation name="RefreshDetached">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshDetached"
style="document"/>
        <wsdl:input>
          <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:message name="CalculateA1SoapIn">
      <wsdl:part name="parameters" element="tns:CalculateA1"/>
    </wsdl:message>
    <wsdl:message name="CalculateA1SoapOut">
      <wsdl:part name="parameters" element="tns:CalculateA1Response"/>
    </wsdl:message>
    <wsdl:message name="CalculateSoapIn">
      <wsdl:part name="parameters" element="tns:Calculate"/>
    </wsdl:message>
    <wsdl:message name="CalculateSoapOut">
      <wsdl:part name="parameters" element="tns:CalculateResponse"/>
    </wsdl:message>
    <wsdl:message name="CalculateWorkbookSoapIn">
      <wsdl:part name="parameters" element="tns:CalculateWorkbook"/>
    </wsdl:message>
    <wsdl:message name="CalculateWorkbookSoapOut">
      <wsdl:part name="parameters" element="tns:CalculateWorkbookResponse"/>
    </wsdl:message>
    <wsdl:message name="CancelRequestSoapIn">
      <wsdl:part name="parameters" element="tns:CancelRequest"/>

```

```

</wsdl:message>
<wsdl:message name="CancelRequestSoapOut">
  <wsdl:part name="parameters" element="tns:CancelRequestResponse"/>
</wsdl:message>
<wsdl:message name="CloseWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:CloseWorkbook"/>
</wsdl:message>
<wsdl:message name="CloseWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:CloseWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="EnsureWorkbookModelSoapIn">
  <wsdl:part name="parameters" element="tns:EnsureWorkbookModel"/>
</wsdl:message>
<wsdl:message name="EnsureWorkbookModelSoapOut">
  <wsdl:part name="parameters" element="tns:EnsureWorkbookModelResponse"/>
</wsdl:message>
<wsdl:message name="GetApiVersionSoapIn">
  <wsdl:part name="parameters" element="tns:GetApiVersion"/>
</wsdl:message>
<wsdl:message name="GetApiVersionSoapOut">
  <wsdl:part name="parameters" element="tns:GetApiVersionResponse"/>
</wsdl:message>
<wsdl:message name="GetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:GetCellA1"/>
</wsdl:message>
<wsdl:message name="GetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:GetCellA1Response"/>
</wsdl:message>
<wsdl:message name="GetCellSoapIn">
  <wsdl:part name="parameters" element="tns:GetCell"/>
</wsdl:message>
<wsdl:message name="GetCellSoapOut">
  <wsdl:part name="parameters" element="tns:GetCellResponse"/>
</wsdl:message>
<wsdl:message name="GetChartImageUrlSoapIn">
  <wsdl:part name="parameters" element="tns:GetChartImageUrl"/>
</wsdl:message>
<wsdl:message name="GetChartImageUrlSoapOut">
  <wsdl:part name="parameters" element="tns:GetChartImageUrlResponse"/>
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNames"/>
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNamesResponse"/>
</wsdl:message>
<wsdl:message name="GetRangeA1SoapIn">
  <wsdl:part name="parameters" element="tns:GetRangeA1"/>
</wsdl:message>
<wsdl:message name="GetRangeA1SoapOut">
  <wsdl:part name="parameters" element="tns:GetRangeA1Response"/>
</wsdl:message>
<wsdl:message name="GetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:GetRange"/>
</wsdl:message>
<wsdl:message name="GetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:GetRangeResponse"/>
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapIn">
  <wsdl:part name="parameters" element="tns:GetSessionInformation"/>
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapOut">
  <wsdl:part name="parameters" element="tns:GetSessionInformationResponse"/>
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetSheetNames"/>
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetSheetNamesResponse"/>

```

```

</wsdl:message>
<wsdl:message name="GetWorkbookConnectionsSoapIn">
  <wsdl:part name="parameters" element="tns:GetWorkbookConnections"/>
</wsdl:message>
<wsdl:message name="GetWorkbookConnectionsSoapOut">
  <wsdl:part name="parameters" element="tns:GetWorkbookConnectionsResponse"/>
</wsdl:message>
<wsdl:message name="GetWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:GetWorkbook"/>
</wsdl:message>
<wsdl:message name="GetWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:GetWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="NewWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:NewWorkbook"/>
</wsdl:message>
<wsdl:message name="NewWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:NewWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookExSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbookEx"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookExSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookExResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookForEditingSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbookForEditing"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookForEditingSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookForEditingResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbook"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="RefreshDetachedSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshDetached"/>
</wsdl:message>
<wsdl:message name="RefreshDetachedSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshDetachedResponse"/>
</wsdl:message>
<wsdl:message name="RefreshExSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshEx"/>
</wsdl:message>
<wsdl:message name="RefreshExSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshExResponse"/>
</wsdl:message>
<wsdl:message name="RefreshSoapIn">
  <wsdl:part name="parameters" element="tns:Refresh"/>
</wsdl:message>
<wsdl:message name="RefreshSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopyDetachedSoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyDetached"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopyDetachedSoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyDetachedResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopySoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopy"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopySoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbook"/>

```

```

</wsdl:message>
<wsdl:message name="SaveWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="SetCalculationOptionsSoapIn">
  <wsdl:part name="parameters" element="tns:SetCalculationOptions"/>
</wsdl:message>
<wsdl:message name="SetCalculationOptionsSoapOut">
  <wsdl:part name="parameters" element="tns:SetCalculationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetCellA1"/>
</wsdl:message>
<wsdl:message name="SetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetCellA1Response"/>
</wsdl:message>
<wsdl:message name="SetCellSoapIn">
  <wsdl:part name="parameters" element="tns:SetCell"/>
</wsdl:message>
<wsdl:message name="SetCellSoapOut">
  <wsdl:part name="parameters" element="tns:SetCellResponse"/>
</wsdl:message>
<wsdl:message name="SetParametersSoapIn">
  <wsdl:part name="parameters" element="tns:SetParameters"/>
</wsdl:message>
<wsdl:message name="SetParametersSoapOut">
  <wsdl:part name="parameters" element="tns:SetParametersResponse"/>
</wsdl:message>
<wsdl:message name="SetRangeA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetRangeA1"/>
</wsdl:message>
<wsdl:message name="SetRangeA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeA1Response"/>
</wsdl:message>
<wsdl:message name="SetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:SetRange"/>
</wsdl:message>
<wsdl:message name="SetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeResponse"/>
</wsdl:message>
</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 Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016

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

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

[<1> Section 2.2.9.1](#): Office SharePoint Server 2007 does not support this error code.

[<2> Section 2.2.9.2](#): Excel Services in Microsoft Office SharePoint Server 2010 and Excel Services in Microsoft Office SharePoint Server 2013 can use any of the values specified in section [2.2.9.1](#).

[<3> Section 3.1.2.1](#): Excel Calculation Services 2007, Microsoft Excel Calculation Services 2010 and Microsoft Excel Calculation Services 2013 time out **sessions (2)** that are not **edit sessions** after a period of inactivity. This period of time is configurable.

[<4> Section 3.1.2.2](#): Excel Services in Microsoft Office SharePoint Server 2007, Excel Services 2010 and Excel Services 2013 time out requests after a period of inactivity. This period of time is configurable.

[<5> Section 3.1.2.2](#): In Excel Calculation Services 2007, Excel Calculation Services 2010 and Excel Calculation Services 2013, only some operations time out. Other operations can exceed the allotted time.

[<6> Section 3.1.4.9](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the **sheet (1)** that was the **active sheet** when the **workbook** was opened.

[<7> Section 3.1.4.13](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

[<8> Section 3.1.4.18](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<9> Section 3.1.4.18](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the **site** when none is specified in the request.

[<10> Section 3.1.4.18](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<11> Section 3.1.4.18](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the request.

[<12> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [\[MSDN-UFES\]](#).

[<13> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [MSDN-UFES].

[<14> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<15> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

[<16> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<17> Section 3.1.4.19](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

[<18> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [MSDN-UFES].

[<19> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [MSDN-UFES].

[<20> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<21> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

[<22> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

[<23> Section 3.1.4.21](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

[<24> Section 3.1.4.30](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

[<25> Section 3.1.4.32](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 set the values sent by the protocol client and do not return an error if **rangeValues** contains values in only one dimension (a single **column** or a single **row**), and if the first **worksheet cell** of **rangeCoordinates** falls on a workbook parameter.

[<26> Section 3.1.4.33](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

[<27> Section 3.1.4.33](#): Excel Services 2007, Excel Services 2010 and Excel Services 2013 set the values sent by the protocol client and do not return an error if **rangeValues** contains values in only one dimension (a single column or a single row), and if the first worksheet cell of **rangeCoordinates** falls on a workbook parameter.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

Section	Description	Revision class
3.1.4.20.2.1 OpenWorkbookEx	Update prescriptive language for clarity.	Major

9 Index

A

Abstract data model
[server](#) 41
[Applicability](#) 20
[ArrayOfAnyType complex type](#) 22
[ArrayOfStatus complex type](#) 22
[ArrayOfString complex type](#) 23
[Attribute groups](#) 27
[Attributes](#) 27

C

[Capability negotiation](#) 20
[CellError simple type](#) 25
[Change tracking](#) 177
[Common data structures](#) 28
 [faults](#) 28
[Common list of subcodes - fault](#) 37
[Complex types](#) 22
 [ArrayOfAnyType](#) 22
 [ArrayOfStatus](#) 22
 [ArrayOfString](#) 23
 [RangeCoordinates](#) 23
 [Status](#) 23
 [TypeExporter](#) 24

D

Data model - abstract
[server](#) 41

E

Events
 [local - server](#) 130
 [timer - server](#) 130
Examples
 [get the value of a worksheet cell](#) 131
 [overview](#) 131
 [refresh data and retrieve a workbook example](#) 134

F

Faults
 [common list of subcodes](#) 37
[Fields - vendor-extensible](#) 20
[Full WSDL](#) 148

G

[Get the value of a worksheet cell example](#) 131
[Glossary](#) 13
[Groups](#) 27

I

[Implementer - security considerations](#) 147
[Index of security parameters](#) 147
[Informative references](#) 19
Initialization

[server](#) 41
[Introduction](#) 13

L

[List of subcodes - common data structures](#) 28
Local events
 [server](#) 130

M

Message processing
 [server](#) 41
Messages
 [ArrayOfAnyType complex type](#) 22
 [ArrayOfStatus complex type](#) 22
 [ArrayOfString complex type](#) 23
 [attribute groups](#) 27
 [attributes](#) 27
 [CellError simple type](#) 25
 [common data structures](#) 28
 [common list of subcodes - fault](#) 37
 [complex types](#) 22
 [elements](#) 22
 [enumerated](#) 21
 [faults](#) 28
 [groups](#) 27
 [namespaces](#) 21
 [RangeCoordinates complex type](#) 23
 [RangeReference simple type](#) 25
 [SaveOptions simple type](#) 26
 [SheetName simple type](#) 26
 [simple types](#) 24
 [Status complex type](#) 23
 [syntax](#) 21
 [transport](#) 21
 [TypeExporter complex type](#) 24
 [WorkbookType simple type](#) 27

N

[Namespaces](#) 21
[Normative references](#) 17

O

Operations
 [Calculate](#) 43
 [CalculateA1](#) 45
 [CalculateWorkbook](#) 48
 [CancelRequest](#) 51
 [CloseWorkbook](#) 53
 [EnsureWorkbookModel](#) 55
 [GetApiVersion](#) 58
 [GetCell](#) 60
 [GetCellA1](#) 62
 [GetChartImageUrl](#) 65
 [GetPublishedItemNames](#) 68
 [GetRange](#) 71
 [GetRangeA1](#) 74
 [GetSessionInformation](#) 77

- [GetSheetNames](#) 80
- [GetWorkbook](#) 84
- [GetWorkbookConnections](#) 86
- [NewWorkbook](#) 88
- [OpenWorkbook](#) 90
- [OpenWorkbookEx](#) 94
- [OpenWorkbookForEditing](#) 96
- [Refresh](#) 99
- [RefreshDetached](#) 101
- [RefreshEx](#) 104
- [SaveWorkbook](#) 106
- [SaveWorkbookCopy](#) 109
- [SaveWorkbookCopyDetached](#) 111
- [SetCalculationOptions](#) 113
- [SetCell](#) 116
- [SetCellA1](#) 119
- [SetParameters](#) 122
- [SetRange](#) 125
- [SetRangeA1](#) 128
- Overview ([synopsis](#)) 19
 - [published items returned](#) 19
 - [results returned](#) 19

P

- [Parameters - security index](#) 147
- [Preconditions](#) 20
- [Prerequisites](#) 20
- [Product behavior](#) 175
- Protocol Details
 - [overview](#) 40

R

- [RangeCoordinates complex type](#) 23
- [RangeReference simple type](#) 25
- [References](#) 17
 - [informative](#) 19
 - [normative](#) 17
- [Refresh data and retrieve a workbook example](#) 134
- [Relationship to other protocols](#) 20
- Request timeout
 - [timers](#) 41
- [Results returned](#) 19

S

- [SaveOptions simple type](#) 26
- Security
 - [implementer considerations](#) 147
 - [parameter index](#) 147
- Sequencing rules
 - [server](#) 41
- Server
 - [abstract data model](#) 41
 - [Calculate operation](#) 43
 - [CalculateA1 operation](#) 45
 - [CalculateWorkbook operation](#) 48
 - [CancelRequest operation](#) 51
 - [CloseWorkbook operation](#) 53
 - [EnsureWorkbookModel operation](#) 55
 - [GetApiVersion operation](#) 58
 - [GetCell operation](#) 60
 - [GetCellA1 operation](#) 62
 - [GetChartImageUrl operation](#) 65
 - [GetPublishedItemNames operation](#) 68
 - [GetRange operation](#) 71
 - [GetRangeA1 operation](#) 74
 - [GetSessionInformation operation](#) 77
 - [GetSheetNames operation](#) 80
 - [GetWorkbook operation](#) 84
 - [GetWorkbookConnections operation](#) 86
 - [initialization](#) 41
 - [local events](#) 130
 - [message processing](#) 41
 - [NewWorkbook operation](#) 88
 - [OpenWorkbook operation](#) 90
 - [OpenWorkbookEx operation](#) 94
 - [OpenWorkbookForEditing operation](#) 96
 - [Refresh operation](#) 99
 - [RefreshDetached operation](#) 101
 - [RefreshEx operation](#) 104
 - [SaveWorkbook operation](#) 106
 - [SaveWorkbookCopy operation](#) 109
 - [SaveWorkbookCopyDetached operation](#) 111
 - [sequencing rules](#) 41
 - [SetCalculationOptions operation](#) 113
 - [SetCell operation](#) 116
 - [SetCellA1 operation](#) 119
 - [SetParameters operation](#) 122
 - [SetRange operation](#) 125
 - [SetRangeA1 operation](#) 128
 - [timer events](#) 130
- Session timeout
 - [timers](#) 41
- [SheetName simple type](#) 26
- [Simple types](#) 24
 - [CellError](#) 25
 - [RangeReference](#) 25
 - [SaveOptions](#) 26
 - [SheetName](#) 26
 - [WorkbookType](#) 27
- [Standards assignments](#) 20
- [Status complex type](#) 23
- Syntax
 - [messages - overview](#) 21

T

- Timer events
 - [server](#) 130
- Timers
 - [request timeout](#) 41
 - [session timeout](#) 41
- [Tracking changes](#) 177
- [Transport](#) 21
- [TypeExporter complex type](#) 24
- Types
 - [complex](#) 22
 - [simple](#) 24

V

- [Vendor-extensible fields](#) 20
- [Versioning](#) 20

W

- [WorkbookType simple type](#) 27
- [WSDL](#) 148

