

[MS-ESP]: Excel Web Services Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0		Revised and edited technical content
10/06/2008	1.01		Revised and edited technical content
12/12/2008	1.02		Revised and edited technical content
07/13/2009	1.03	Major	Revised and edited the technical content
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
01/20/2012	3.0	Major	Significantly changed the technical content.
04/11/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Preliminary

Table of Contents

1 Introduction	12
1.1 Glossary	12
1.2 References	14
1.2.1 Normative References	14
1.2.2 Informative References	15
1.3 Protocol Overview (Synopsis)	15
1.3.1 Published Items and Results Returned by this Protocol	16
1.4 Relationship to Other Protocols	16
1.5 Prerequisites/Preconditions	16
1.6 Applicability Statement	17
1.7 Versioning and Capability Negotiation	17
1.8 Vendor-Extensible Fields	17
1.9 Standards Assignments	17
2 Messages	18
2.1 Transport	18
2.2 Common Message Syntax	18
2.2.1 Namespaces	18
2.2.2 Messages	18
2.2.3 Elements	19
2.2.4 Complex Types	19
2.2.4.1 ArrayOfAnyType	19
2.2.4.2 ArrayOfStatus	19
2.2.4.3 RangeCoordinates	20
2.2.4.4 Status	20
2.2.4.5 TypeExporter	21
2.2.5 Simple Types	21
2.2.5.1 CellError	21
2.2.5.2 RangeReference	22
2.2.5.3 SheetName	22
2.2.5.4 WorkbookType	23
2.2.6 Attributes	23
2.2.7 Groups	23
2.2.8 Attribute Groups	23
2.2.9 Faults	23
2.2.9.1 List of Subcodes	24
2.2.9.2 Common List of Subcodes	33
3 Protocol Details	34
3.1 Server Details	35
3.1.1 Abstract Data Model	35
3.1.2 Timers	35
3.1.2.1 Session Timeout	35
3.1.2.2 Request Timeout	35
3.1.3 Initialization	35
3.1.4 Message Processing Events and Sequencing Rules	35
3.1.4.1 Calculate	37
3.1.4.1.1 Messages	38
3.1.4.1.1.1 CalculateSoapIn	38
3.1.4.1.1.2 CalculateSoapOut	38

3.1.4.1.2	Elements.....	38
3.1.4.1.2.1	Calculate	38
3.1.4.1.2.2	CalculateResponse.....	39
3.1.4.1.3	Complex Types	39
3.1.4.1.4	Simple Types.....	39
3.1.4.1.5	Attributes.....	39
3.1.4.1.6	Groups.....	39
3.1.4.1.7	Attribute Groups	39
3.1.4.2	CalculateA1.....	39
3.1.4.2.1	Messages	40
3.1.4.2.1.1	CalculateA1SoapIn.....	40
3.1.4.2.1.2	CalculateA1SoapOut	40
3.1.4.2.2	Elements.....	41
3.1.4.2.2.1	CalculateA1	41
3.1.4.2.2.2	CalculateA1Response	41
3.1.4.2.3	Complex Types	41
3.1.4.2.4	Simple Types.....	41
3.1.4.2.5	Attributes.....	41
3.1.4.2.6	Groups.....	42
3.1.4.2.7	Attribute Groups	42
3.1.4.3	CalculateWorkbook	42
3.1.4.3.1	Messages	42
3.1.4.3.1.1	CalculateWorkbookSoapIn	42
3.1.4.3.1.2	CalculateWorkbookSoapOut	43
3.1.4.3.2	Elements.....	43
3.1.4.3.2.1	CalculateWorkbook	43
3.1.4.3.2.2	CalculateWorkbookResponse.....	43
3.1.4.3.3	Complex Types	43
3.1.4.3.4	Simple Types.....	44
3.1.4.3.4.1	CalculateType	44
3.1.4.3.5	Attributes.....	44
3.1.4.3.6	Groups.....	44
3.1.4.3.7	Attribute Groups	44
3.1.4.4	CancelRequest	44
3.1.4.4.1	Messages	45
3.1.4.4.1.1	CancelRequestSoapIn	45
3.1.4.4.1.2	CancelRequestSoapOut	45
3.1.4.4.2	Elements.....	45
3.1.4.4.2.1	CancelRequest	45
3.1.4.4.2.2	CancelRequestResponse	45
3.1.4.4.3	Complex Types	46
3.1.4.4.4	Simple Types.....	46
3.1.4.4.5	Attributes.....	46
3.1.4.4.6	Groups.....	46
3.1.4.4.7	Attribute Groups	46
3.1.4.5	CloseWorkbook	46
3.1.4.5.1	Messages	46
3.1.4.5.1.1	CloseWorkbookSoapIn	46
3.1.4.5.1.2	CloseWorkbookSoapOut	47
3.1.4.5.2	Elements.....	47
3.1.4.5.2.1	CloseWorkbook	47
3.1.4.5.2.2	CloseWorkbookResponse	47
3.1.4.5.3	Complex Types	47

3.1.4.5.4	Simple Types	48
3.1.4.5.5	Attributes	48
3.1.4.5.6	Groups	48
3.1.4.5.7	Attribute Groups	48
3.1.4.6	GetApiVersion	48
3.1.4.6.1	Messages	48
3.1.4.6.1.1	GetApiVersionSoapIn	48
3.1.4.6.1.2	GetApiVersionSoapOut	48
3.1.4.6.2	Elements	49
3.1.4.6.2.1	GetApiVersion	49
3.1.4.6.2.2	GetApiVersionResponse	49
3.1.4.6.3	Complex Types	49
3.1.4.6.4	Simple Types	49
3.1.4.6.5	Attributes	49
3.1.4.6.6	Groups	49
3.1.4.6.7	Attribute Groups	49
3.1.4.7	GetCell	49
3.1.4.7.1	Messages	50
3.1.4.7.1.1	GetCellSoapIn	50
3.1.4.7.1.2	GetCellSoapOut	50
3.1.4.7.2	Elements	51
3.1.4.7.2.1	GetCell	51
3.1.4.7.2.2	GetCellResponse	51
3.1.4.7.3	Complex Types	52
3.1.4.7.4	Simple Types	52
3.1.4.7.5	Attributes	52
3.1.4.7.6	Groups	52
3.1.4.7.7	Attribute Groups	52
3.1.4.8	GetCellA1	52
3.1.4.8.1	Messages	53
3.1.4.8.1.1	GetCellA1SoapIn	53
3.1.4.8.1.2	GetCellA1SoapOut	53
3.1.4.8.2	Elements	53
3.1.4.8.2.1	GetCellA1	53
3.1.4.8.2.2	GetCellA1Response	54
3.1.4.8.3	Complex Types	54
3.1.4.8.4	Simple Types	54
3.1.4.8.5	Attributes	54
3.1.4.8.6	Groups	54
3.1.4.8.7	Attribute Groups	54
3.1.4.9	GetChartImageUrl	54
3.1.4.9.1	Messages	55
3.1.4.9.1.1	GetChartImageUrlSoapIn	55
3.1.4.9.1.2	GetChartImageUrlSoapOut	55
3.1.4.9.2	Elements	55
3.1.4.9.2.1	GetChartImageUrl	55
3.1.4.9.2.2	GetChartImageUrlResponse	56
3.1.4.9.3	Complex Types	56
3.1.4.9.3.1	Size	56
3.1.4.9.4	Simple Types	56
3.1.4.9.5	Attributes	57
3.1.4.9.6	Groups	57
3.1.4.9.7	Attribute Groups	57

3.1.4.10	GetPublishedItemNames	57
3.1.4.10.1	Messages	57
3.1.4.10.1.1	GetPublishedItemNamesSoapIn	57
3.1.4.10.1.2	GetPublishedItemNamesSoapOut	57
3.1.4.10.2	Elements	58
3.1.4.10.2.1	GetPublishedItemNames	58
3.1.4.10.2.2	GetPublishedItemNamesResponse	58
3.1.4.10.3	Complex Types	58
3.1.4.10.3.1	ArrayOfWorkbookItem	58
3.1.4.10.3.2	WorkbookItem	59
3.1.4.10.4	Simple Types	59
3.1.4.10.4.1	ItemType	59
3.1.4.10.5	Attributes	59
3.1.4.10.6	Groups	59
3.1.4.10.7	Attribute Groups	59
3.1.4.11	GetRange	60
3.1.4.11.1	Messages	61
3.1.4.11.1.1	GetRangeSoapIn	61
3.1.4.11.1.2	GetRangeSoapOut	61
3.1.4.11.2	Elements	61
3.1.4.11.2.1	GetRange	61
3.1.4.11.2.2	GetRangeResponse	62
3.1.4.11.3	Complex Types	62
3.1.4.11.4	Simple Types	62
3.1.4.11.5	Attributes	62
3.1.4.11.6	Groups	62
3.1.4.11.7	Attribute Groups	62
3.1.4.12	GetRangeA1	62
3.1.4.12.1	Messages	63
3.1.4.12.1.1	GetRangeA1SoapIn	63
3.1.4.12.1.2	GetRangeA1SoapOut	64
3.1.4.12.2	Elements	64
3.1.4.12.2.1	GetRangeA1	64
3.1.4.12.2.2	GetRangeA1Response	64
3.1.4.12.3	Complex Types	65
3.1.4.12.4	Simple Types	65
3.1.4.12.5	Attributes	65
3.1.4.12.6	Groups	65
3.1.4.12.7	Attribute Groups	65
3.1.4.13	GetSessionInformation	65
3.1.4.13.1	Messages	66
3.1.4.13.1.1	GetSessionInformationSoapIn	66
3.1.4.13.1.2	GetSessionInformationSoapOut	66
3.1.4.13.2	Elements	66
3.1.4.13.2.1	GetSessionInformation	66
3.1.4.13.2.2	GetSessionInformationResponse	66
3.1.4.13.3	Complex Types	67
3.1.4.13.4	Simple Types	67
3.1.4.13.5	Attributes	67
3.1.4.13.6	Groups	67
3.1.4.13.7	Attribute Groups	67
3.1.4.14	GetSheetNames	67
3.1.4.14.1	Messages	68

3.1.4.14.1.1	GetSheetNamesSoapIn.....	68
3.1.4.14.1.2	GetSheetNamesSoapOut.....	68
3.1.4.14.2	Elements.....	68
3.1.4.14.2.1	GetSheetNames.....	68
3.1.4.14.2.2	GetSheetNamesResponse.....	68
3.1.4.14.3	Complex Types.....	69
3.1.4.14.3.1	ArrayOfSheetInfo.....	69
3.1.4.14.3.2	SheetInfo.....	69
3.1.4.14.4	Simple Types.....	69
3.1.4.14.4.1	SheetType.....	69
3.1.4.14.4.2	SheetVisibility.....	70
3.1.4.14.5	Attributes.....	70
3.1.4.14.6	Groups.....	70
3.1.4.14.7	Attribute Groups.....	70
3.1.4.15	GetWorkbook.....	70
3.1.4.15.1	Messages.....	71
3.1.4.15.1.1	GetWorkbookSoapIn.....	71
3.1.4.15.1.2	GetWorkbookSoapOut.....	71
3.1.4.15.2	Elements.....	71
3.1.4.15.2.1	GetWorkbook.....	71
3.1.4.15.2.2	GetWorkbookResponse.....	72
3.1.4.15.3	Complex Types.....	72
3.1.4.15.4	Simple Types.....	72
3.1.4.15.5	Attributes.....	72
3.1.4.15.6	Groups.....	72
3.1.4.15.7	Attribute Groups.....	72
3.1.4.16	NewWorkbook.....	72
3.1.4.16.1	Messages.....	73
3.1.4.16.1.1	NewWorkbookSoapIn.....	73
3.1.4.16.1.2	NewWorkbookSoapOut.....	73
3.1.4.16.2	Elements.....	73
3.1.4.16.2.1	NewWorkbook.....	73
3.1.4.16.2.2	NewWorkbookResponse.....	74
3.1.4.16.3	Complex Types.....	74
3.1.4.16.4	Simple Types.....	74
3.1.4.16.5	Attributes.....	74
3.1.4.16.6	Groups.....	74
3.1.4.16.7	Attribute Groups.....	74
3.1.4.17	OpenWorkbook.....	75
3.1.4.17.1	Messages.....	76
3.1.4.17.1.1	OpenWorkbookSoapIn.....	76
3.1.4.17.1.2	OpenWorkbookSoapOut.....	76
3.1.4.17.2	Elements.....	76
3.1.4.17.2.1	OpenWorkbook.....	76
3.1.4.17.2.2	OpenWorkbookResponse.....	77
3.1.4.17.3	Complex Types.....	77
3.1.4.17.4	Simple Types.....	77
3.1.4.17.5	Attributes.....	77
3.1.4.17.6	Groups.....	77
3.1.4.17.7	Attribute Groups.....	77
3.1.4.18	OpenWorkbookForEditing.....	78
3.1.4.18.1	Messages.....	79
3.1.4.18.1.1	OpenWorkbookForEditingSoapIn.....	79

3.1.4.18.1.2	OpenWorkbookForEditingSoapOut.....	79
3.1.4.18.2	Elements.....	79
3.1.4.18.2.1	OpenWorkbookForEditing.....	79
3.1.4.18.2.2	OpenWorkbookForEditingResponse	80
3.1.4.18.3	Complex Types	80
3.1.4.18.4	Simple Types	80
3.1.4.18.5	Attributes.....	80
3.1.4.18.6	Groups.....	80
3.1.4.18.7	Attribute Groups.....	80
3.1.4.19	Refresh.....	81
3.1.4.19.1	Messages	81
3.1.4.19.1.1	RefreshSoapIn.....	81
3.1.4.19.1.2	RefreshSoapOut	81
3.1.4.19.2	Elements.....	81
3.1.4.19.2.1	Refresh	81
3.1.4.19.2.2	RefreshResponse	82
3.1.4.19.3	Complex Types	82
3.1.4.19.4	Simple Types	82
3.1.4.19.5	Attributes.....	82
3.1.4.19.6	Groups.....	82
3.1.4.19.7	Attribute Groups.....	82
3.1.4.20	SaveWorkbook	82
3.1.4.20.1	Messages	83
3.1.4.20.1.1	SaveWorkbookSoapIn	83
3.1.4.20.1.2	SaveWorkbookSoapOut	83
3.1.4.20.2	Elements.....	83
3.1.4.20.2.1	SaveWorkbook	83
3.1.4.20.2.2	SaveWorkbookResponse.....	84
3.1.4.20.3	Complex Types	84
3.1.4.20.4	Simple Types	84
3.1.4.20.5	Attributes.....	84
3.1.4.20.6	Groups.....	84
3.1.4.20.7	Attribute Groups.....	84
3.1.4.21	SaveWorkbookCopy	84
3.1.4.21.1	Messages	85
3.1.4.21.1.1	SaveWorkbookCopySoapIn	85
3.1.4.21.1.2	SaveWorkbookCopySoapOut	85
3.1.4.21.2	Elements.....	86
3.1.4.21.2.1	SaveWorkbookCopy	86
3.1.4.21.2.2	SaveWorkbookCopyResponse.....	86
3.1.4.21.3	Complex Types	86
3.1.4.21.4	Simple Types.....	87
3.1.4.21.4.1	SaveOptions.....	87
3.1.4.21.5	Attributes.....	87
3.1.4.21.6	Groups.....	87
3.1.4.21.7	Attribute Groups.....	87
3.1.4.22	SetCalculationOptions.....	87
3.1.4.22.1	Messages	88
3.1.4.22.1.1	SetCalculationOptionsSoapIn.....	88
3.1.4.22.1.2	SetCalculationOptionsSoapOut	88
3.1.4.22.2	Elements.....	88
3.1.4.22.2.1	SetCalculationOptions	88
3.1.4.22.2.2	SetCalculationOptionsResponse	88

3.1.4.22.3	Complex Types	89
3.1.4.22.4	Simple Types	89
3.1.4.22.4.1	WorkbookCalculation	89
3.1.4.22.5	Attributes	89
3.1.4.22.6	Groups	89
3.1.4.22.7	Attribute Groups	89
3.1.4.23	SetCell	90
3.1.4.23.1	Messages	90
3.1.4.23.1.1	SetCellSoapIn	90
3.1.4.23.1.2	SetCellSoapOut	90
3.1.4.23.2	Elements	91
3.1.4.23.2.1	SetCell	91
3.1.4.23.2.2	SetCellResponse	91
3.1.4.23.3	Complex Types	91
3.1.4.23.4	Simple Types	92
3.1.4.23.5	Attributes	92
3.1.4.23.6	Groups	92
3.1.4.23.7	Attribute Groups	92
3.1.4.24	SetCellA1	92
3.1.4.24.1	Messages	93
3.1.4.24.1.1	SetCellA1SoapIn	93
3.1.4.24.1.2	SetCellA1SoapOut	93
3.1.4.24.2	Elements	93
3.1.4.24.2.1	SetCellA1	93
3.1.4.24.2.2	SetCellA1Response	94
3.1.4.24.3	Complex Types	94
3.1.4.24.4	Simple Types	94
3.1.4.24.5	Attributes	94
3.1.4.24.6	Groups	94
3.1.4.24.7	Attribute Groups	94
3.1.4.25	SetParameters	94
3.1.4.25.1	Messages	95
3.1.4.25.1.1	SetParametersSoapIn	95
3.1.4.25.1.2	SetParametersSoapOut	96
3.1.4.25.2	Elements	96
3.1.4.25.2.1	SetParameters	96
3.1.4.25.2.2	SetParametersResponse	96
3.1.4.25.3	Complex Types	96
3.1.4.25.3.1	ArrayOfParameterInfo	96
3.1.4.25.3.2	ParameterInfo	97
3.1.4.25.3.3	ArrayOfString	97
3.1.4.25.4	Simple Types	97
3.1.4.25.5	Attributes	97
3.1.4.25.6	Groups	97
3.1.4.25.7	Attribute Groups	98
3.1.4.26	SetRange	98
3.1.4.26.1	Messages	99
3.1.4.26.1.1	SetRangeSoapIn	99
3.1.4.26.1.2	SetRangeSoapOut	99
3.1.4.26.2	Elements	99
3.1.4.26.2.1	SetRange	99
3.1.4.26.2.2	SetRangeResponse	100
3.1.4.26.3	Complex Types	100

3.1.4.26.4	Simple Types	100
3.1.4.26.5	Attributes	100
3.1.4.26.6	Groups	100
3.1.4.26.7	Attribute Groups	100
3.1.4.27	SetRangeA1	100
3.1.4.27.1	Messages	101
3.1.4.27.1.1	SetRangeA1SoapIn	101
3.1.4.27.1.2	SetRangeA1SoapOut	101
3.1.4.27.2	Elements	101
3.1.4.27.2.1	SetRangeA1	101
3.1.4.27.2.2	SetRangeA1Response	102
3.1.4.27.3	Complex Types	102
3.1.4.27.4	Simple Types	102
3.1.4.27.5	Attributes	102
3.1.4.27.6	Groups	102
3.1.4.27.7	Attribute Groups	102
3.1.5	Timer Events	103
3.1.6	Other Local Events	103
4	Protocol Examples	104
4.1	Getting the Value of a Worksheet Cell	104
4.2	Refreshing Data and Retrieving a Workbook	107
5	Security	119
5.1	Security Considerations for Implementers	119
5.2	Index of Security Parameters	119
6	Appendix A: Full WSDL	120
7	Appendix B: Product Behavior	146
8	Change Tracking	148
9	Index	149

1 Introduction

This document specifies the Excel Web Services Protocol, which is a Web service protocol that is used to read from and perform basic operations on workbooks hosted on a protocol server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Universal Naming Convention (UNC)

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

active sheet
array formula
AutoFilter
automatic calculation mode
calculate
calculation mode
cell
chart sheet
checked out
column
confirmation
connected data
cube
data connection
data culture
data provider
data table
default filter value
dimension
document library
edit session
farm
filter
formula
grow
Information Rights Management (IRM)
item
level
license key
list
locked
manual calculation mode
manual filter
merged cell
named object

named range
new workbook session
non-contiguous range
number format
OLAP
Open Item permission
permission
PivotTable
PivotTable member
PivotTable page field
published
published item
published items snapshot
published sheet
query
range
refresh
row
session
session identifier
sheet
Simple Object Access Protocol (SOAP)
site
 slicer
snapshot
SOAP action
SOAP body
SOAP fault
table
trusted location
UI culture
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
user-defined function (UDF)
view session
view-only mode
visible
Web service
Web Services Description Language (WSDL)
workbook
workbook file
workbook parameter
worksheet
WSDL message
WSDL operation
XML namespace
XML schema

The following terms are specific to this document:

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

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

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

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

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

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

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

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

[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.ietf.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.ietf.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.ietf.org/rfc/rfc3986.txt>

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

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

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

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

1.2.2 Informative References

[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, "About Cell and Range References", <http://office.microsoft.com/en-us/help/HP051983231033.aspx>

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

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

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

1.3 Protocol Overview (Synopsis)

This protocol 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:

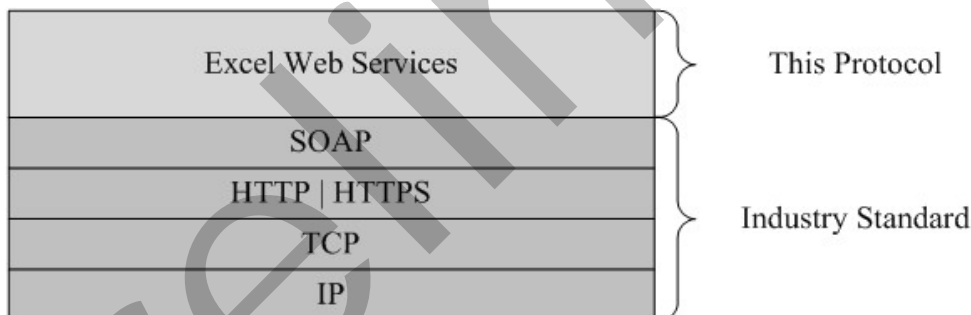


Figure 1: This protocol in relation to other protocols

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

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** by 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
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
tns	http://schemas.microsoft.com/office/excel/server/webservices	
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]

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.
ArrayOfStatus	An array of status information.
RangeCoordinates	Specifies a range on a workbook.
Status	The status information returned by the protocol server.
TypeExporter	MUST NOT be used.

2.2.4.1 ArrayOfAnyType

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

```
<xs:complexType name="ArrayOfAnyType">  
  <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

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">  
  <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 RangeCoordinates

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">
  <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 (2)** index. MUST be greater than or equal to 0 and less than or equal to 1,048,575.

Column: The **column (2)** index. MUST be greater than or equal to 0 and less than or equal to 16,383.

Height: The number of rows (2). 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 (2). MUST be greater than or equal to 0. The sum of **Column** and **Width** MUST NOT be greater than 16,384.

2.2.4.4 Status

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

```
<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>
```

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.

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

Message: The message description.

2.2.4.5 TypeExporter

MUST NOT be used. It is defined as follows.

```
<xs:complexType name="TypeExporter">
  <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 in a workbook, in some of the operations specified by this protocol.
SheetName	A string that is used by the protocol server to find a sheet in a workbook in some of the operations specified in this protocol.
WorkbookType	An enumeration that specifies the form of a requested workbook.

2.2.5.1 CellError

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">
  <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:simpleType>
```

The following table specifies the allowable values for CellError:

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

A string that is used by the protocol server to identify a range on a sheet 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">  
  <xs:restriction base="xs:string"/>  
</xs:simpleType>
```

2.2.5.3 SheetName

A string that is used by the protocol server to find a sheet 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 in the workbook unless specified otherwise in the operation.
- If no sheet 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 is not a **visible** sheet.

This simple type is defined as follows.

```
<xs:simpleType name="SheetName">  
  <xs:restriction base="xs:string"/>  
</xs:simpleType>
```

2.2.5.4 WorkbookType

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">  
  <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 WorkbookType:

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 Faults

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 List of Subcodes

The following table describes the error codes.

Value	Meaning
Blank	MUST NOT be used.
InternalError	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.
ExternalDataRefreshFailedReadOrViewOnly	Connected data refresh failed.
NoAccessToOdcReadOrViewOnly	Connected data refresh failed.
ErrorConnectingToDataSourceReadOrViewOnly	Connected data refresh failed.
InvalidUserNameOrPasswordReadOrViewOnly	Connected data refresh failed.
InvalidUserNameOrPassword	Connected data refresh failed.
ExternalDatatNotAllowedOnTL	Connected data refresh failed.
InvalidUriorLocalConnectionPath	Connected data refresh failed.
DclNotTrusted	Connected data refresh failed.
OdcFileDoesNotExist	Connected data refresh failed.
ErrorAccessingSsofromWfe	Connected data refresh failed.
ErrorAccessingSsoFromEcs	Connected data refresh failed.
SsoError	Connected data refresh failed.
ErrorConnectingToDataSource	Connected data refresh failed.
UnattendedAccountNotConfigured	Connected data refresh failed.
IncorrectAuthUsed	Connected data refresh failed.
ExternalDataRefreshFailed_Details	Connected data refresh failed.
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

Value	Meaning
	list (2).
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 (2) .
DataSetAxisTooBig	The row (2) or column (2) 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.

Value	Meaning
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.
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.

Value	Meaning
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.
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

Value	Meaning
	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.
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 (2) 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.
InsDelCellsArrayFmla	The operation failed because it tried to change part of

Value	Meaning
	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 (2), but the specified range is a whole column (2).
InsDelCellsColumnsForWholeRow	The operation failed because it tried to insert or delete a whole column (2), but the specified range is a whole row (2).
InvalidSheetName	The operation failed because the specified sheet 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 location.
DiscontiguousRangeNotSupported	The operation failed because a non-contiguous range was specified.
RangeRequestAreaExceeded	The requested range was too large.
SheetRangeMismatch	The sheet provided as the sheet argument is not the same as the sheet 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.

Value	Meaning
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.
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 (2) that is not in view-only mode timed out while the user was attempting to convert it to

Value	Meaning
	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.
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.

Value	Meaning
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.
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 [<1>](#) 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**, **present**, and **not null**.

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.

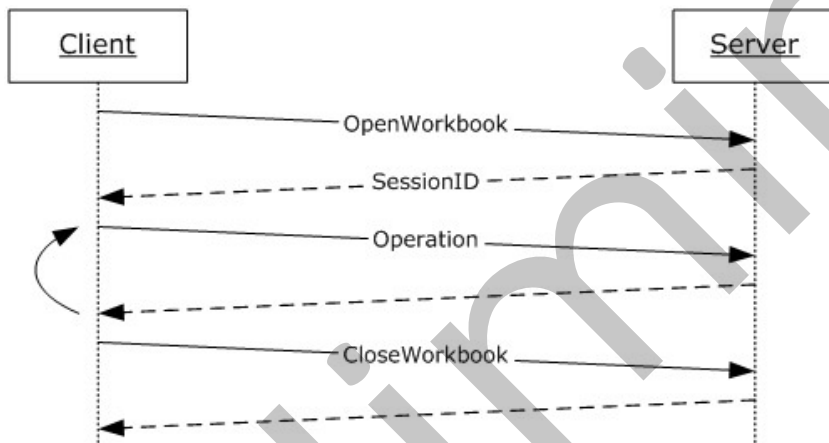


Figure 2: High-level sequence diagram

Session Behavior

View Sessions

A view session is created on a successful **OpenWorkbook** 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 3.1.4.18.2.1). 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 specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

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<2> 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<3> time requests out. When a request times out, the protocol server SHOULD<4> return a SOAP fault response, with the subcode **MaxRequestDurationExceeded**.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

Groups that represent required and optional interfaces are shown in the following tables. Group 1 is the required set of interfaces. Group 2 is an optional set of interfaces<5>.

If one interface from an optional group is implemented, all interfaces in that group MUST be implemented.

The groups form an additive set: Group 1 is required. If one interface from Group 2 is implemented, all interfaces in Group 1 and Group 2 are implemented.

The following table summarizes the list of WSDL operations belonging to Group 1 as defined by this specification.

WSDL 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.
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.
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.
GetRange	This operation is used to retrieve the contents of a named 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.
GetWorkbook	This operation is used to retrieve a workbook file.
OpenWorkbook	This operation is used to create a new view session, giving access to a workbook on the protocol server.
Refresh	This operation is used to refresh connected data on a workbook.
SetCell	This operation is used to set a value to a specific worksheet cell in the workbook.
SetCellA1	This operation is used to set a value to a specific worksheet cell in the 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.

The following table summarizes the list of WSDL operations belonging to Group 2 as defined by this specification.

WSDL operation	Description
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.
GetSheetNames	The operation returns information about the sheets in a workbook.
NewWorkbook	This operation is used to create a new workbook session.
OpenWorkbookForEditing	This operation is used to create a new edit session, giving access to a workbook on the protocol server.
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.
SetCalculationOptions	This operation is used to set the workbook calculation mode .
SetParameters	This operation to sets values of workbook parameters in a workbook.

3.1.4.1 Calculate

This operation is used to calculate a specific range in the workbook and is defined as follows.

```
<wsdl:operation name="Calculate">
  <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 doesn't 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 [2.2.4.3](#), and the **sheetName** is not empty, the whole sheet, 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

Subcode
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.1.1 Messages

3.1.4.1.1.1 CalculateSoapIn

The requested WSDL message for a **Calculate WSDL operation**.

The **SOAP action** value is:

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

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

3.1.4.1.1.2 CalculateSoapOut

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

The SOAP action value is:

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

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

3.1.4.1.2 Elements

3.1.4.1.2.1 Calculate

The input data for a **Calculate** WSDL operation.

```
<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>
```

sessionId: A **session identifier (2)**. 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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

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

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

3.1.4.1.2.2 CalculateResponse

The result data for a **Calculate** WSDL operation.

```
<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>
```

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 and is defined as follows.

```
<wsdl:operation name="CalculateA1">
  <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 doesn't 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, 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

3.1.4.2.1.1 CalculateA1SoapIn

The requested WSDL message for a **CalculateA1** WSDL operation.

The SOAP action value is:

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

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

3.1.4.2.1.2 CalculateA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.2.2 Elements

3.1.4.2.2.1 CalculateA1

The input data for a **CalculateA1** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the sheet 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 result data for a **CalculateA1** WSDL operation.

```
<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>
```

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 defined as follows.

```
<wsdl:operation name="CalculateWorkbook">
  <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
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.3.1 Messages

3.1.4.3.1.1 CalculateWorkbookSoapIn

The requested WSDL message for a **CalculateWorkbook** WSDL operation.

The SOAP action value is:

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

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

3.1.4.3.1.2 CalculateWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.3.2 Elements

3.1.4.3.2.1 CalculateWorkbook

The input data for a **CalculateWorkbook** WSDL operation.

```
<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>
```

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**, **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 result data for a **CalculateWorkbook** WSDL operation.

```
<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>
```

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

3.1.4.3.4.1 CalculateType

The type of calculation to be executed.

```
<xs:simpleType name="CalculateType">  
  <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 CalculateType:

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 and is defined as follows.

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

3.1.4.4.1.1 CancelRequestSoapIn

The requested WSDL message for a **CancelRequest** WSDL operation.

The SOAP action value is:

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

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

3.1.4.4.1.2 CancelRequestSoapOut

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

The SOAP action value is:

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

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

3.1.4.4.2 Elements

3.1.4.4.2.1 CancelRequest

The input data for a **CancelRequest** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, or but MAY instead be obtained in any other way.

3.1.4.4.2.2 CancelRequestResponse

The result data for a **CancelRequest** WSDL operation.

```
<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>
```

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 and is defined as follows.

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

3.1.4.5.1.1 CloseWorkbookSoapIn

The requested WSDL message for a **CloseWorkbook** WSDL operation.

The SOAP action value is:

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

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

3.1.4.5.1.2 CloseWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.5.2 Elements

3.1.4.5.2.1 CloseWorkbook

The input data for a **CloseWorkbook** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, or but MAY instead be obtained in any other way.

3.1.4.5.2.2 CloseWorkbookResponse

The result data for a **CloseWorkbook** WSDL operation.

```
<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>
```

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 GetApiVersion

This operation is used to get the version of the protocol server and is defined as follows.

```
<wsdl:operation name="GetApiVersion">
  <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.

3.1.4.6.1 Messages

3.1.4.6.1.1 GetApiVersionSoapIn

The requested WSDL message for a **GetApiVersion** WSDL operation.

The SOAP action value is:

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

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

3.1.4.6.1.2 GetApiVersionSoapOut

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

The SOAP action value is:

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

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

3.1.4.6.2 Elements

3.1.4.6.2.1 GetApiVersion

The input data for a **GetApiVersion** WSDL operation.

```
<xs:element name="GetApiVersion">  
  <xs:complexType/>  
</xs:element>
```

3.1.4.6.2.2 GetApiVersionResponse

The result data for a **GetApiVersion** WSDL operation.

```
<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>
```

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

status: MUST NOT be present.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 GetCell

This operation is used to retrieve the contents of a specific cell in the workbook and is defined as follows.

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

```

    <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 (2), column (2) on the sheet specified by **sheetName**, specify the location of the sheet cells to retrieve.
2. If **formatted** is **false**, the sheet 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 sheet cell returned as the **GetCellResult** element MUST be type **string**, and MUST be formatted using a **number format** according to how the sheet 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

3.1.4.7.1 Messages

3.1.4.7.1.1 GetCellSoapIn

The requested WSDL message for a **GetCell** WSDL operation.

The SOAP action value is:

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

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

3.1.4.7.1.2 GetCellSoapOut

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

The SOAP action value is:

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

3.1.4.7.2 Elements

3.1.4.7.2.1 GetCell

The input data for a **GetCell** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the cell. See section [2.2.5.3](#). MUST be present. MUST NOT be empty.

row: The row (2) index of the cell. MUST be greater than or equal to 0 and less than or equal to 1,048,575.

column: The column (2) 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.

3.1.4.7.2.2 GetCellResponse

The result data for a **GetCell** WSDL operation.

```
<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>
```

GetCellResult: The returned cell value.

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

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 Groupos

None.

3.1.4.8 GetCellA1

This operation is used to retrieve the contents of a specific cell in the workbook and is defined as follows.

```
<wsdl:operation name="GetCellA1">
  <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:

- The **rangeName** specifies the location of the sheet cell to retrieve. The sheet that contains the sheet cell to retrieve is determined in the following manner.
- If **rangeName** specifies a sheet, that is the sheet which MUST be used.
- Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section [2.2.5.3](#).
- If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY [<6>](#) use any sheet in the workbook.
- If **formatted** is false, the sheet 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](#)).
- If **formatted** is true, the sheet cell returned as the **GetCellA1Result** element MUST be type **string**, and MUST be number formatted according to how the sheet 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

3.1.4.8.1 Messages

3.1.4.8.1.1 GetCellA1SoapIn

The requested WSDL message for a **GetCellA1** WSDL operation.

The SOAP action value is:

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

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

3.1.4.8.1.2 GetCellA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.8.2 Elements

3.1.4.8.2.1 GetCellA1

The input data for a **GetCellA1** WSDL operation.

```
<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>

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the cell. See section [2.2.5.3](#) 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.

3.1.4.8.2.2 GetCellA1Response

The result data for a **GetCellA1** WSDL operation.

```
<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>
```

GetCellA1Result: 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 GetChartImageUrl

This operation returns the URL of a chart in a workbook and is defined as follows.

```

<wsdl:operation name="GetChartImageUrl">
  <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.

3.1.4.9.1 Messages

3.1.4.9.1.1 GetChartImageUrlSoapIn

The requested WSDL message for a **GetChartImageUrl** WSDL operation.

The SOAP action value is:

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

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

3.1.4.9.1.2 GetChartImageUrlSoapOut

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

The SOAP action value is:

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

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

3.1.4.9.2 Elements

3.1.4.9.2.1 GetChartImageUrl

The input data for a **GetChartImageUrl** WSDL operation.

```

<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:element>

```

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

size: The size of the chart image as defined in section [3.1.4.9.3.1](#). If not present, the original chart size will be used.

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

3.1.4.9.2.2 GetChartImageUrlResponse

The result data for a **GetChartImageUrl** WSDL operation.

```
<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>
```

GetChartImageUrlResult: The URL of the requested chart.

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

3.1.4.9.3 Complex Types

3.1.4.9.3.1 Size

Specifies the size of a chart image defined as follows.

```
<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>
```

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.

3.1.4.9.4 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 GetPublishedItemNames

This operation returns a list of the published items in a workbook and is defined as follows.

```
<wsdl:operation name="GetPublishedItemNames">
  <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.

3.1.4.10.1 Messages

3.1.4.10.1.1 GetPublishedItemNamesSoapIn

The requested WSDL message for a **GetPublishedItemNames** WSDL operation.

The SOAP action value is:

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

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

3.1.4.10.1.2 GetPublishedItemNamesSoapOut

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

The SOAP action value is:

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

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

3.1.4.10.2 Elements

3.1.4.10.2.1 GetPublishedItemNames

The input data for a **GetPublishedItemNames** WSDL operation.

```
<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>
```

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

3.1.4.10.2.2 GetPublishedItemNamesResponse

The result data for a **GetPublishedItemNames** WSDL operation.

```
<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>
```

GetPublishedItemNamesResult: An array of **WorkbookItem** elements as specified in section [3.1.4.10.3.1](#). MUST be present.

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

3.1.4.10.3 Complex Types

3.1.4.10.3.1 ArrayOfWorkbookItem

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

```
<xs:complexType name="ArrayOfWorkbookItem">
  <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 [3.1.4.10.3.2](#).

3.1.4.10.3.2 WorkbookItem

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

```
<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>
```

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 [3.1.4.10.4.1](#).

3.1.4.10.4 Simple Types

3.1.4.10.4.1 ItemType

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

```
<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>
```

The following table specifies the allowable values for ItemType:

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 (2).
NamedRange	Specifies that the published item is a named range.

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 GetRange

This operation is used to retrieve the contents of a specific range in the workbook and is defined as follows.

```
<wsdl:operation name="GetRange">
  <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 specified by **sheetName**, specify the location of the sheet cells to retrieve.
2. Each element inside the **GetRangeResult** element MUST be of type **ArrayOfAnyType**. Each element inside **GetRangeResult** is a row (2) of sheet cells returned from the protocol server. The number of rows (2) returned MUST be equal to the number of rows (2) requested in the **rangeCoordinates** element.
3. Each element inside each of the **ArrayOfAnyType** type nested inside **GetRangeResult** MUST be a single workbook sheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of columns (2) requested in the **rangeCoordinates** element.
4. If **formatted** is false, all sheet 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 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

3.1.4.11.1 Messages

3.1.4.11.1.1 GetRangeSoapIn

The requested WSDL message for a **GetRange** WSDL operation.

The SOAP action value is:

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

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

3.1.4.11.1.2 GetRangeSoapOut

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

The SOAP action value is:

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

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

3.1.4.11.2 Elements

3.1.4.11.2.1 GetRange

The input data for a **GetRange** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the range. See section [2.2.5.3](#) 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 [2.2.4.3](#).
- The product of **Height** and **Width** MUST NOT be greater than 1,000,000.

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

3.1.4.11.2.2 GetRangeResponse

The result data for a **GetRange** WSDL operation.

```
<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>
```

GetRangeResult: The returned range value.

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

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 GetRangeA1

This operation is used to retrieve the contents of a named range in the workbook and is defined as follows.

```
<wsdl:operation name="GetRangeA1">
  <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 to retrieve. The sheet that contains the sheet cells to retrieve is determined in the following manner:
 - If **rangeName** specifies a sheet, that is the sheet which MUST be used.
 - Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section [2.2.5.3](#). If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY [<7>](#) use any sheet in the workbook.
- Each element inside the **GetRangeA1Result** element MUST be of type **ArrayOfAnyType**. Each element inside the **GetRangeA1Result** is a row (2) of sheet cells returned from the protocol server. The number of rows (2) MUST be equal to the rows (2) in the named range specified by **rangeName**.
- Each element inside each of the **ArrayOfAnyType** type nested inside **GetRangeA1Result** MUST be a single workbook sheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of columns (2) in the named range specified by **rangeName**.
- If **formatted** is false, all sheet 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 cells returned as part of the **GetRangeA1Result** element MUST be type **string**, and MUST be number formatted according to how the sheet 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

3.1.4.12.1 Messages

3.1.4.12.1.1 GetRangeA1SoapIn

The requested WSDL message for a **GetRangeA1** WSDL operation.

The SOAP action value is:

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

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

3.1.4.12.1.2 GetRangeA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.12.2 Elements

3.1.4.12.2.1 GetRangeA1

The input data for a **GetRangeA1** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

sheetName: The name of the sheet on which the **rangeName** is defined. See section [2.2.5.3](#) 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.

3.1.4.12.2.2 GetRangeA1Response

The result data for a **GetRangeA1** WSDL operation.

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

GetRangeA1Result: The returned range value.

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

3.1.4.12.3 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 GetSessionInformation

This operation retrieves information about a session (2) on the protocol server and is defined as follows.

```
<wsdl:operation name="GetSessionInformation">
  <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.

3.1.4.13.1 Messages

3.1.4.13.1.1 GetSessionInformationSoapIn

The requested WSDL message for a **GetSessionInformation** WSDL operation.

The SOAP action value is:

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

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

3.1.4.13.1.2 GetSessionInformationSoapOut

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

The SOAP action value is:

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

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

3.1.4.13.2 Elements

3.1.4.13.2.1 GetSessionInformation

The input data for a **GetSessionInformation** WSDL operation.

```
<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>
```

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

3.1.4.13.2.2 GetSessionInformationResponse

The result data for a **GetSessionInformation** WSDL operation.

```
<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>

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](#).

3.1.4.13.3 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 GetSheetNames

The operation returns information about the sheets in a workbook defined as follows.

```
<wsdl:operation name="GetSheetNames">
  <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 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.

3.1.4.14.1 Messages

3.1.4.14.1.1 GetSheetNamesSoapIn

The requested WSDL message for a **GetSheetNames** WSDL operation.

The SOAP action value is:

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

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

3.1.4.14.1.2 GetSheetNamesSoapOut

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

The SOAP action value is:

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

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

3.1.4.14.2 Elements

3.1.4.14.2.1 GetSheetNames

The input data for a **GetSheetNames** WSDL operation.

```
<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>
```

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

3.1.4.14.2.2 GetSheetNamesResponse

The result data for a **GetSheetNames** WSDL operation.

```
<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>
```

GetSheetNamesResult: An array of **SheetInfo** elements as specified in section [3.1.4.14.3.1](#). MUST be present.

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

3.1.4.14.3 Complex Types

3.1.4.14.3.1 ArrayOfSheetInfo

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

```
<xs:complexType name="ArrayOfSheetInfo">
  <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 [3.1.4.14.3.2](#).

3.1.4.14.3.2 SheetInfo

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

```
<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>
```

Name: The name of the sheet. MUST be present.

Type: The type of the sheet as specified in section [3.1.4.14.4.1](#).

Visibility: The visible condition of the sheet as specified in section [3.1.4.14.4.2](#).

3.1.4.14.4 Simple Types

3.1.4.14.4.1 SheetType

Specifies the type of a sheet and is defined as follows.

```
<xs:simpleType name="SheetType">
  <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 SheetType:

Value	Meaning
Worksheet	A worksheet.
Chartsheet	A chart sheet .

3.1.4.14.4.2 SheetVisibility

Specifies the visible condition of a sheet and is defined as follows.

```
<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>
```

The following table specifies the allowable values for SheetVisibility:

Value	Meaning
Visible	Specifies that the sheet is visible.
Hidden	Specifies that the sheet is not visible and cannot be displayed.
VeryHidden	Specifies that the sheet is not visible and cannot be displayed.

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 GetWorkbook

This operation is used to retrieve a workbook file and is defined as follows.

```
<wsdl:operation name="GetWorkbook">  
  <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 an SOAP fault with subcode **EcsSnapshotAccessDenied**.
- If the protocol server cannot retrieve the workbook file in the form specified by **workbookType**, the protocol server returns an 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

3.1.4.15.1 Messages

3.1.4.15.1.1 GetWorkbookSoapIn

The requested WSDL message for a **GetWorkbook** WSDL operation.

The SOAP action value is:

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

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

3.1.4.15.1.2 GetWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.15.2 Elements

3.1.4.15.2.1 GetWorkbook

The input data for a **GetWorkbook** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

workbookType: The form of the requested workbook as specified in section [2.2.5.4](#).

3.1.4.15.2 GetWorkbookResponse

The result data for a **GetWorkbook** WSDL operation.

```
<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>
```

GetWorkbookResult: A workbook file. MUST be present.

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

3.1.4.15.3 Complex Types

None.

3.1.4.15.4 Simple Types

None.

3.1.4.15.5 Attributes

None.

3.1.4.15.6 Groups

None.

3.1.4.15.7 Attribute Groups

None.

3.1.4.16 NewWorkbook

This operation is used to create a new workbook session and is defined as follows.

```
<wsdl:operation name="NewWorkbook">
  <wsdl:input message="tns:NewWorkbookSoapIn"/>
  <wsdl:output message="tns:NewWorkbookSoapOut"/>
</wsdl:operation>
```


</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 (2) 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.

3.1.4.16.1 Messages

3.1.4.16.1.1 NewWorkbookSoapIn

The requested WSDL message for a **NewWorkbook** WSDL operation.

The SOAP action value is:

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

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

3.1.4.16.1.2 NewWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.16.2 Elements

3.1.4.16.2.1 NewWorkbook

The input data for a **NewWorkbook** WSDL operation.

```
<xs:element name="NewWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="path" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

    <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 (2)** 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.

3.1.4.16.2.2 NewWorkbookResponse

The result data for a **NewWorkbook** WSDL operation.

```

<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>

```

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

status: The status information returned by the protocol server.

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 OpenWorkbook

This operation is used to create a new view session, giving access to a workbook on the protocol server and is defined as follows.

```
<wsdl:operation name="OpenWorkbook">
  <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 [\[MS-XLSB\]](#), [\[ISO/IEC-29500-1\]](#), [\[MS-OFFMACRO2\]](#), or [\[ODF\]](#) but MAY be in any other format.
 2. If it is in the [\[ISO/IEC-29500-1\]](#) format:
 1. It MAY [<12>](#) contain any valid SpreadsheetML records in that format.
 2. If a **odcFile** attribute of a **connection** element ([\[ISO/IEC-29500-1\]](#), 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 ([\[ISO/IEC-29500-1\]](#), 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 (2) 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
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError
WorkbookNotSupported

3.1.4.17.1 Messages

3.1.4.17.1.1 OpenWorkbookSoapIn

The requested WSDL message for a **OpenWorkbook** WSDL operation.

The SOAP action value is:

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

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

3.1.4.17.1.2 OpenWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.17.2 Elements

3.1.4.17.2.1 OpenWorkbook

The input data for a **OpenWorkbook** WSDL operation.

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

```
<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.

3.1.4.17.2 OpenWorkbookResponse

The result data for a **OpenWorkbook** WSDL operation.

```
<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>
```

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

status: The status information returned by the protocol server.

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 OpenWorkbookForEditing

This operation is used to create a new edit session, giving access to a workbook on the protocol server and is defined as follows.

```
<wsdl:operation name="OpenWorkbookForEditing">
  <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 [\[MS-XLSB\]](#), [\[ISO/IEC-29500-1\]](#), [\[MS-OFFMACRO2\]](#), or [\[ODF\]](#) but MAY be in any other format.
 2. If it is in the [\[ISO/IEC-29500-1\]](#) format:
 1. It MAY [<18>](#) contain any valid SpreadsheetML records in that format.
 2. If a **odcFile** attribute of a **connection** element ([\[ISO/IEC-29500-1\]](#), 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 ([\[ISO/IEC-29500-1\]](#), 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 (2) 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
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError

3.1.4.18.1 Messages

3.1.4.18.1.1 OpenWorkbookForEditingSoapIn

The requested WSDL message for a **OpenWorkbookForEditing** WSDL operation.

The SOAP action value is:

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

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

3.1.4.18.1.2 OpenWorkbookForEditingSoapOut

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

The SOAP action value is:

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

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

3.1.4.18.2 Elements

3.1.4.18.2.1 OpenWorkbookForEditing

The input data for a **OpenWorkbookForEditing** WSDL operation.

```
<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: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.

3.1.4.18.2.2 OpenWorkbookForEditingResponse

The result data for a **OpenWorkbookForEditing** WSDL operation.

```
<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>
```

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

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

3.1.4.18.3 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 Refresh

This operation is used to refresh connected data on a workbook and is defined as follows.

```
<wsdl:operation name="Refresh">
  <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 (2)** in the workbook if any exist. If **connectionName** is present and is not empty, the protocol server refreshes the specified data connections (2) 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.

3.1.4.19.1 Messages

3.1.4.19.1.1 RefreshSoapIn

The requested WSDL message for a **Refresh** WSDL operation.

The SOAP action value is:

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

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

3.1.4.19.1.2 RefreshSoapOut

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

The SOAP action value is:

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

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

3.1.4.19.2 Elements

3.1.4.19.2.1 Refresh

The input data for a **Refresh** WSDL operation.

```
<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: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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

connectionName: A name of a data connection (2) in the workbook. MUST NOT be longer than 255 characters.

3.1.4.19.2.2 RefreshResponse

The result data for a **Refresh** WSDL operation.

```
<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>
```

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

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 SaveWorkbook

This operation is used to save a workbook associated with an open session (2) to the same location from which it was opened. This operation is defined as follows.

```
<wsdl:operation name="SaveWorkbook">
  <wsdl:input message="tns:SaveWorkbookSoapIn"/>
  <wsdl:output message="tns:SaveWorkbookSoapOut"/>
</wsdl:operation>
```

</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

3.1.4.20.1 Messages

3.1.4.20.1.1 SaveWorkbookSoapIn

The requested WSDL message for a **SaveWorkbook** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SaveWorkbook** element.

3.1.4.20.1.2 SaveWorkbookSoapOut

The response WSDL message for a **SaveWorkbook** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SaveWorkbookResponse** element.

3.1.4.20.2 Elements

3.1.4.20.2.1 SaveWorkbook

The input data for a **SaveWorkbook** WSDL operation.

```

<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>

```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

3.1.4.20.2 SaveWorkbookResponse

The result data for a **SaveWorkbook** WSDL operation.

```

<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>

```

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

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 SaveWorkbookCopy

This operation is used to save a workbook associated with an open session (2) to a specified location, and in a specified format. This operation is defined as follows.

```

<wsdl:operation name="SaveWorkbookCopy">
  <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 an 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

3.1.4.21.1 Messages

3.1.4.21.1.1 SaveWorkbookCopySoapIn

The requested WSDL message for a **SaveWorkbookCopy** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SaveWorkbookCopy** element.

3.1.4.21.1.2 SaveWorkbookCopySoapOut

The response WSDL message for a **SaveWorkbookCopy** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SaveWorkbookCopyResponse** element.

3.1.4.21.2 Elements

3.1.4.21.2.1 SaveWorkbookCopy

The input data for a **SaveWorkbookCopy** WSDL operation.

```
<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>
```

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**, **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 [2.2.5.4](#).

saveOptions: A combination of 0 or more values of **SaveOptions** (section [3.1.4.21.4.1](#)). MUST be present.

3.1.4.21.2.2 SaveWorkbookCopyResponse

The result data for a **SaveWorkbookCopy** WSDL operation.

```
<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>
```

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

3.1.4.21.4.1 SaveOptions

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

```
<xs:simpleType name="SaveOptions">
  <xs:list>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="AllowOverwrite"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:list>
</xs:simpleType>
```

The following table specifies the allowable values for SaveOptions:

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.

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 SetCalculationOptions

This operation is used to set the workbook calculation mode and is defined as follows.

```
<wsdl:operation name="SetCalculationOptions">
  <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.

3.1.4.22.1 Messages

3.1.4.22.1.1 SetCalculationOptionsSoapIn

The requested WSDL message for a **SetCalculationOptions** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetCalculationOptions** element.

3.1.4.22.1.2 SetCalculationOptionsSoapOut

The response WSDL message for a **SetCalculationOptions** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetCalculationOptionsResponse** element.

3.1.4.22.2 Elements

3.1.4.22.2.1 SetCalculationOptions

The input data for a **SetCalculationOptions** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

calcType: The calculation mode value as specified in **WorkbookCalculation**.

3.1.4.22.2.2 SetCalculationOptionsResponse

The result data for a **SetCalculationOptions** WSDL operation.


```

<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>

```

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

3.1.4.22.3 Complex Types

None.

3.1.4.22.4 Simple Types

3.1.4.22.4.1 WorkbookCalculation

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">
  <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 WorkbookCalculation:

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.

3.1.4.22.5 Attributes

None.

3.1.4.22.6 Groups

None.

3.1.4.22.7 Attribute Groups

None.

3.1.4.23 SetCell

This operation is used to set a value to a specific worksheet cell in the workbook. This operation is defined as follows.

```
<wsdl:operation name="SetCell">
  <wsdl:input message="tns:SetCellSoapIn"/>
  <wsdl:output message="tns:SetCellSoapOut"/>
</wsdl:operation>
```

1. The row (2) and column (2) fields on the sheets 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

3.1.4.23.1 Messages

3.1.4.23.1.1 SetCellSoapIn

The requested WSDL message for a **SetCell** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetCell** element.

3.1.4.23.1.2 SetCellSoapOut

The response WSDL message for a **SetCell** WSDL operation.

The SOAP action value is:

The SOAP body contains a **SetCellResponse** element.

3.1.4.23.2 Elements

3.1.4.23.2.1 SetCell

The input data for a **SetCell** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the worksheet cell to set. See section [2.2.5.3](#). MUST be present. MUST NOT be empty.

row: The row (2) 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 (2) 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.

3.1.4.23.2.2 SetCellResponse

The result data for a **SetCell** WSDL operation.

```
<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>
```

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

3.1.4.23.3 Complex Types

None.

3.1.4.23.4 Simple Types

None.

3.1.4.23.5 Attributes

None.

3.1.4.23.6 Groups

None.

3.1.4.23.7 Attribute Groups

None.

3.1.4.24 SetCellA1

This operation is used to set a value to a specific worksheet cell in the workbook. This operation is defined as follows.

```
<wsdl:operation name="SetCellA1">
  <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 that contains the worksheet cell that will be set is determined in the following manner:
 - If **rangeName** specifies a sheet, that sheet MUST be used.
 - Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section [2.2.5.3](#).
 - If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY [<24>](#) use any sheet in the workbook.
2. If **sheetName** is empty and **rangeName** specifies a sheet, as defined in section [2.2.5.2](#), the protocol server MUST use the sheet 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

3.1.4.24.1 Messages

3.1.4.24.1.1 SetCellA1SoapIn

The requested WSDL message for a **SetCellA1** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1
```

The SOAP body contains a **SetCellA1** element.

3.1.4.24.1.2 SetCellA1SoapOut

The response WSDL message for a **SetCellA1** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1
```

The SOAP body contains a **SetCellA1Response** element.

3.1.4.24.2 Elements

3.1.4.24.2.1 SetCellA1

The input data for a **SetCellA1** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the worksheet cell to set. See section [2.2.5.3](#). 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.

3.1.4.24.2 SetCellA1Response

The result data for a **SetCellA1** WSDL operation.

```
<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>
```

status: An array of status information 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 SetParameters

This operation sets values of workbook parameters in a workbook. This operation is defined as follows.

```
<wsdl:operation name="SetParameters">
  <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 [3.1.4.25.3.2](#)) 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

3.1.4.25.1 Messages

3.1.4.25.1.1 SetParametersSoapIn

The requested WSDL message for a **SetParameters** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetParameters** element.

3.1.4.25.1.2 SetParametersSoapOut

The response WSDL message for a **SetParameters** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetParametersResponse** element.

3.1.4.25.2 Elements

3.1.4.25.2.1 SetParameters

The input data for a **SetParameters** WSDL operation.

```
<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>
```

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

parameters: An array of **ParameterInfo** elements (section [3.1.4.25.3.2](#)) to be set in the workbook, as specified in section [3.1.4.25.3.1](#). MUST be present.

3.1.4.25.2.2 SetParametersResponse

The result data for a **SetParameters** WSDL operation.

```
<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>
```

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

3.1.4.25.3 Complex Types

3.1.4.25.3.1 ArrayOfParameterInfo

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">
  <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 [3.1.4.25.3.2](#).

3.1.4.25.3.2 ParameterInfo

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">
  <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 [3.1.4.25.3.3](#).

3.1.4.25.3.3 ArrayOfString

An array of string values defined as follows.

```
<xs:complexType name="ArrayOfString">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

string: A string value. MUST NOT exceed 1024 characters.

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 SetRange

This operation is used to set values to a range of worksheet cells on a workbook. This operation is defined as follows.

```
<wsdl:operation name="SetRange">
  <wsdl:input message="tns:SetRangeSoapIn"/>
  <wsdl:output message="tns:SetRangeSoapOut"/>
</wsdl:operation>
```

1. The **rangeCoordinates** on the sheet 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 (2) of worksheet cells. The number of rows (2) in **rangeValues** MUST be equal to the number of rows (2) 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 (2) 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

3.1.4.26.1 Messages

3.1.4.26.1.1 SetRangeSoapIn

The requested WSDL message for a **SetRange** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetRange** element.

3.1.4.26.1.2 SetRangeSoapOut

The response WSDL message for a **SetRange** WSDL operation.

The SOAP action value is:

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

The SOAP body contains a **SetRangeResponse** element.

3.1.4.26.2 Elements

3.1.4.26.2.1 SetRange

The input data for a **SetRange** WSDL operation.

```
<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>
```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the range to set. See section [2.2.5.3](#). 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.

3.1.4.26.2.2 SetRangeResponse

The result data for a **SetRange** WSDL operation.

```
<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>
```

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 SetRangeA1

This operation is used to set values to a range of worksheet cells on a workbook. This operation is defined as follows.

```
<wsdl:operation name="SetRangeA1">
  <wsdl:input message="tns:SetRangeA1SoapIn"/>
  <wsdl:output message="tns:SetRangeA1SoapOut"/>
</wsdl:operation>
```

1. The sheet that contains the range that will be set is determined in the following manner:
 - If **rangeName** specifies a sheet, that sheet MUST be used.
 - Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section [2.2.5.3](#).
 - If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY [<26>](#) use any sheet in the workbook.

2. If **sheetName** is empty and **rangeName** specifies a sheet, as defined in section [2.2.5.2](#), the sheet 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 (2) of worksheet cells. The number of rows (2) in **rangeValues** MUST be equal to the number of rows (2) 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 (2) 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.

3.1.4.27.1 Messages

3.1.4.27.1.1 SetRangeA1SoapIn

The requested WSDL message for a **SetRangeA1** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1
```

The SOAP body contains a **SetRangeA1** element.

3.1.4.27.1.2 SetRangeA1SoapOut

The response WSDL message for a **SetRangeA1** WSDL operation.

The SOAP action value is:

```
http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1
```

The SOAP body contains a **SetRangeA1Response** element.

3.1.4.27.2 Elements

3.1.4.27.2.1 SetRangeA1

The input data for a **SetRangeA1** WSDL operation.

```
<x: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>

```

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**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.

sheetName: The name of the sheet that contains the range to set. See section [2.2.5.3](#). 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.

3.1.4.27.2 SetRangeA1Response

The result data for a **SetRangeA1** WSDL operation.

```

<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>

```

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

3.1.4.27.3 Complex Types

None.

3.1.4.27.4 Simple Types

None.

3.1.4.27.5 Attributes

None.

3.1.4.27.6 Groups

None.

3.1.4.27.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

Preliminary

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.

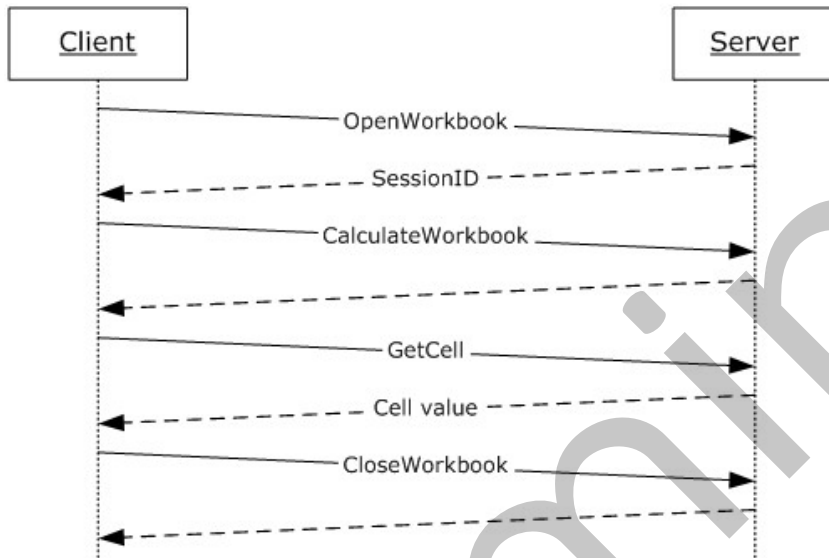


Figure 3: Getting the value of a worksheet cell example

The order of operations, including the relevant **SOAP** requests and responses for each step, is as follows:

- 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>
```



```
</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>
```

- 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>
```

- 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>

```

- 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.

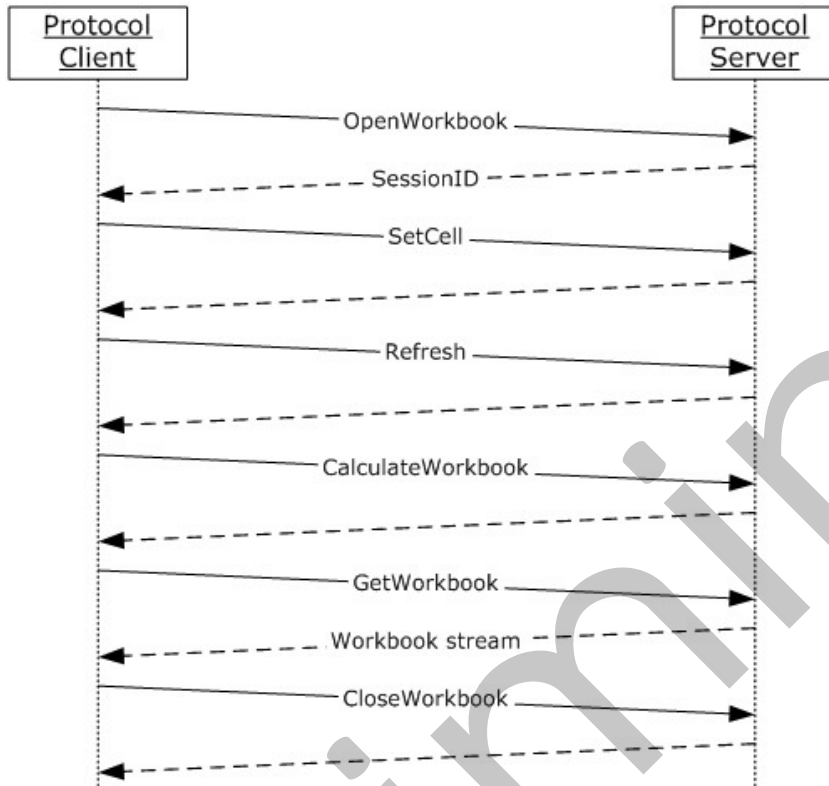


Figure 4: Refresh data and retrieve workbook example

The order of operations, including the relevant SOAP requests and responses for each step, is as follows:

- 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>
```

```
</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>
```

- 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>
```

- 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>
```

- 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

```
<?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>
```



```

12b3RUYWJsZXMvX3JlbHMvcG12b3RUYWJsZTEueG1sLnJlbHNQSwECLQAUAAYACAAAACEAojvW8o4EAAA5QQAAHgAAAA
AAAAAAAAAACPEQAeGwvcG12b3RUYWJsZXMvX3JlbHMvcG12b3RUYWJsZTEueG1sUEsBAi0AFAAGAAgAAAAhAERJdzrnrAAAAIgEA
ABQAAAAAAAAAAAAAAAAAWRYAAHhsL3NoYXJlZFN0cm1uZ3MueG1sUEsBAi0AFAAGAAgAAAAhABc1P8r0BgAATCMAABMAA
AAAAAAAAAAAAAAAAAchcAAHhsL3RoZW11L3RoZW11MS54bWxQSwECLQAUAAYACAAAACEAs6QaM6YCAACYBgAAGAAAAAAAAA
AAAAAAAAACXHGAAeGwvd29ya3NoZWV0cy9zaGVldDEueG1sUEsBAi0AFAAGAAgAAAAhAJQj1tyoAgAANGYAAA0AAAAAAAA
AAAAAAAAAAcyEAAHhsL3NoeWxlcycy54bWxQSwECLQAUAAYACAAAACEAGpZbxGkBAACMAGAAgAAAAAAAAAAAAAAAAABGJAAA
eGwvd29ya3NoZWV0cy9zaGVldDMueG1sUEsBAi0AFAAGAAgAAAAhAJCVG1tV8JwAARooBACCAAAAAAAAAAAAAAAAAA5SUA
HhsL3Bpdm90Q2FjAGUvcG12b3RDYWN0ZUR1ZmluaXRpb24xLnhtbFBLAQITABQABGAIATAAAIQATKrfqzAEAAngCAASAA
AAAAAAAAAAAAAAAAAKZNAAB4bC9jb25uZWNoaW9ucy54bWxQSwECLQAUAAYACAAAACEALmnZ2SkBAAA0BAAAJwAAAAAAAAA
AAAAAAAAACiTWAAeGwvcHJpbmR1c1NldHRpbmdzL3ByaW50ZXJtZXRoZWV0aW5nczMuYmluUEsBAi0AFAAGAAgAAAAhAC5p2dkp
AQANAQAACCAAAAAAAAAAAAAAAAAAEFEAAHhsL3ByaW50ZXJtZXRoZWV0aW5ncy9wcm1udGVyU2V0dGluZ3MyLmJpb1BLAQITa
BQABGAIATAAAIQAUadnZKQEAAQEEAAAnAAAAAAAAAAAAAAAAAH5SAAB4bC9wcm1udGVyU2V0dGluZ3MvchJpbmR1c1NldH
RpbmdzMS5iaW50QSwECLQAUAAYACAAAACEAvYriI5AAAADbAAAAEwAAAAAAAAAAAAAAAAADsUwAAy3VzdG9tWG1sL2L10ZW0
zLnhtbFBLAQITABQABGAIATAAAIQAMxBqSwAAACgBAAAEAAAAAAAAAAAAAAAAANVUAABjdxN0b21YbWwvX3JlbHMvaXR1
bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEa/MCo8MAAAAOAQAAGHAAAAAAAAAAAAAAAAADUVQAAy3VzdG9tWG1sL19yZ
WxzL2L10ZW0zLnhtbC5yZWxzUEsBAi0AFAAGAAgAAAAhAFYJyLDAAAKAEAB4AAAAAAAAAAAAAAAAAA01YAAGN1c3RvbV
htbC9fcMvscy9pdGVtMi54bWwvcVsc1BLAQITABQABGAIATAAAIQB0Pz16wgAAACgBAAAEAAAAAAAAAAAAAAAAANJXAAB
jdxN0b21YbWwvX3JlbHMvaXR1bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEABk/130sBAACBAAQAAAAAAAAAAAAAAAA
AADQWAAAAG9jUHJvcHMvY29yZS54bWxQSwECLQAUAAYACAAAACEakFZEKpQBAABHAAEAAAAAAAAAAAAAAAAABSwwAAZ
G9jUHJvcHMvYXN0b21YbWwvX3JlbHMvaXR1bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEAAjJrAABjdXN0b21YbWwvX3JlbHMvaXR1
bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEA2jZYA7YAADJAAAAGAAAAAAAAAAAAAAAAACjbaAAy3VzdG9tWG1sL2L10ZW0zLnht
1sUEsBAi0AFAAGAAgAAAAhANLlyxi4AAAAyQAAABGAAAAAAAAAAAAAAAAAAt20AAGN1c3RvbVhtbC9pdGVtUHJvcHMvLnht
bFBLAQITABQABGAIATAAAIQAE3E8qItwAAAMKAAAAyAAAAAAAAAAAAAAAAAM1uAABjdXN0b21YbWwvX3JlbHMvaXR1bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEAksWz0hgBAAAKAgAAEwAAAAAAAAAAAAAAAAADibwAAy3VzdG9tWG1sL2L10ZW0zLnhtbFBLB
QYAAAAIwAjAMJAAABTcQAAAAA=</GetWorkbookResult>
</GetWorkbookResponse>
</soap:Body>
</soap:Envelope>

```

- 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.V23.494r81aHjSSiziqJhY1U2B590.5.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>

Preliminary

5 Security

5.1 Security Considerations for Implementers

There are security risks associated with exposing session identifiers (2). 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.

Preliminary

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL and schema are provided in this appendix.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
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"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <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:schema>
  </wsdl:types>

```



```

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

```

```

        <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: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 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:sequence>
    </xs:complexType>
</xs:element>
<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: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: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: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="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="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="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="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="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="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="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="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="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="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="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="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="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="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="SetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:SetRange" />
</wsdl:message>
<wsdl:message name="SetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeResponse" />
</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="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="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="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="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="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="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="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="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="GetChartImageUrlSoapIn">
  <wsdl:part name="parameters" element="tns:GetChartImageUrl" />
</wsdl:message>
<wsdl:message name="GetChartImageUrlSoapOut">
  <wsdl:part name="parameters" element="tns:GetChartImageUrlResponse" />
</wsdl:message>
<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="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: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: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="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: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/GetSessionInformatio
n" 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: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/OpenWorkbookForEdit
ing" 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:input>

```

```

        <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="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: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:binding>
</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 Preview

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 2.2.9.2](#): Excel Services 2010 can use any of the values specified in section [2.2.9.1](#).

<2> [Section 3.1.2.1](#): Excel Calculation Services 2007 and Excel Calculation Services 2010 time out sessions (2) that are not edit sessions after a period of inactivity. This period of time is configurable.

<3> [Section 3.1.2.2](#): Excel Services 2007 and Excel Services 2010 time out requests after a period of inactivity. This period of time is configurable.

<4> [Section 3.1.2.2](#): In Excel Calculation Services 2007 and Excel Calculation Services 2010, only some operations time out. Other operations can exceed the allotted time.

<5> [Section 3.1.4](#): Group 1 is supported in Excel Services 2007 and Excel Services 2010. Group 2 is supported in Excel Services 2010.

<6> [Section 3.1.4.8](#): Excel Services 2007 and Excel Services 2010 use the sheet that was the **active sheet** when the workbook was opened.

<7> [Section 3.1.4.12](#): Excel Services 2007 and Excel Services 2010 use the sheet that was the active sheet when the workbook was opened.

<8> [Section 3.1.4.16](#): Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<9> [Section 3.1.4.16](#): Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the request.

<10> [Section 3.1.4.16](#): Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<11> [Section 3.1.4.16](#): Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the request.

<12> [Section 3.1.4.17](#): Excel Services 2007 and Excel Services 2010 have exceptions as noted in [\[MSDN-UFES\]](#).

<13> [Section 3.1.4.17:](#) Excel Services 2007 and Excel Services 2010 have exceptions as noted in [\[MSDN-UFES\]](#).

<14> [Section 3.1.4.17:](#) Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<15> [Section 3.1.4.17:](#) Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the current request.

<16> [Section 3.1.4.17:](#) Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<17> [Section 3.1.4.17:](#) Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the current request.

<18> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 have exceptions as noted in [\[MSDN-UFES\]](#).

<19> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 have exceptions as noted in [\[MSDN-UFES\]](#).

<20> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<21> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the current request.

<22> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 attempt to fall back to a supported language when an unsupported language is requested.

<23> [Section 3.1.4.18:](#) Excel Services 2007 and Excel Services 2010 use the language set for the site (2) when none is specified in the current request.

<24> [Section 3.1.4.24:](#) Excel Services 2007 and Excel Services 2010 use the sheet that was the active sheet when the workbook was opened.

<25> [Section 3.1.4.26:](#) Excel Services 2007 and Excel Services 2010 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 (2) or a single row (2)), and if the first worksheet cell of **rangeCoordinates** falls on a workbook parameter.

<26> [Section 3.1.4.27:](#) Excel Services 2007 and Excel Services 2010 use the sheet that was the active sheet when the workbook was opened.

<27> [Section 3.1.4.27:](#) Excel Services 2007 and Excel Services 2010 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

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

Preliminary

9 Index

A

Abstract data model
 [server](#) 35
[Applicability](#) 17
[ArrayOfAnyType complex type](#) 19
[ArrayOfStatus complex type](#) 19
[Attribute groups](#) 23
[Attributes](#) 23

C

[Capability negotiation](#) 17
[CellError simple type](#) 21
[Change tracking](#) 148
Client
 [overview](#) 34
[Common list of subcodes - fault](#) 33
[Complex types](#) 19
 [ArrayOfAnyType](#) 19
 [ArrayOfStatus](#) 19
 [RangeCoordinates](#) 20
 [Status](#) 20
 [TypeExporter](#) 21

D

Data model - abstract
 [server](#) 35

E

Events
 [local - server](#) 103
 [timer - server](#) 103
Examples
 [get the value of a worksheet cell](#) 104
 [overview](#) 104
 [refresh data and retrieve a workbook example](#)
 107

F

[Faults](#) 23
 [common list of subcodes](#) 33
 [list of subcodes](#) 24
[Fields - vendor-extensible](#) 17
[Full WSDL](#) 120

G

[Get the value of a worksheet cell example](#) 104
[Glossary](#) 12
[Groups](#) 23

I

[Implementer - security considerations](#) 119
[Index of security parameters](#) 119

[Informative references](#) 15
Initialization
 [server](#) 35
[Introduction](#) 12

L

[List of subcodes - fault](#) 24
Local events
 [server](#) 103

M

Message processing
 [server](#) 35
Messages
 [ArrayOfAnyType complex type](#) 19
 [ArrayOfStatus complex type](#) 19
 [attribute groups](#) 23
 [attributes](#) 23
 [CellError simple type](#) 21
 [common list of subcodes - fault](#) 33
 [complex types](#) 19
 [elements](#) 19
 [enumerated](#) 18
 [faults](#) 23
 [groups](#) 23
 [list of subcodes - fault](#) 24
 [namespaces](#) 18
 [RangeCoordinates complex type](#) 20
 [RangeReference simple type](#) 22
 [SheetName simple type](#) 22
 [simple types](#) 21
 [Status complex type](#) 20
 [syntax](#) 18
 [transport](#) 18
 [TypeExporter complex type](#) 21
 [WorkbookType simple type](#) 23

N

[Namespaces](#) 18
[Normative references](#) 14

O

Operations
 [Calculate](#) 37
 [CalculateA1](#) 39
 [CalculateWorkbook](#) 42
 [CancelRequest](#) 44
 [CloseWorkbook](#) 46
 [GetApiVersion](#) 48
 [GetCell](#) 49
 [GetCellA1](#) 52
 [GetChartImageUrl](#) 54
 [GetPublishedItemNames](#) 57
 [GetRange](#) 60
 [GetRangeA1](#) 62

[GetSessionInformation](#) 65
[GetSheetNames](#) 67
[GetWorkbook](#) 70
[NewWorkbook](#) 72
[OpenWorkbook](#) 75
[OpenWorkbookForEditing](#) 78
[Refresh](#) 81
[SaveWorkbook](#) 82
[SaveWorkbookCopy](#) 84
[SetCalculationOptions](#) 87
[SetCell](#) 90
[SetCellA1](#) 92
[SetParameters](#) 94
[SetRange](#) 98
[SetRangeA1](#) 100
[Overview \(synopsis\)](#) 15
[published items returned](#) 16
[results returned](#) 16

P

[Parameters - security index](#) 119
[Preconditions](#) 16
[Prerequisites](#) 16
[Product behavior](#) 146

R

[RangeCoordinates complex type](#) 20
[RangeReference simple type](#) 22
[References](#) 14
[informative](#) 15
[normative](#) 14
[Refresh data and retrieve a workbook example](#) 107
[Relationship to other protocols](#) 16
[Request timeout](#)
[timers](#) 35
[Results returned](#) 16

S

Security
[implementer considerations](#) 119
[parameter index](#) 119
Sequencing rules
[server](#) 35
Server
[abstract data model](#) 35
[Calculate operation](#) 37
[CalculateA1 operation](#) 39
[CalculateWorkbook operation](#) 42
[CancelRequest operation](#) 44
[CloseWorkbook operation](#) 46
[GetApiVersion operation](#) 48
[GetCell operation](#) 49
[GetCellA1 operation](#) 52
[GetChartImageUrl operation](#) 54
[GetPublishedItemNames operation](#) 57
[GetRange operation](#) 60
[GetRangeA1 operation](#) 62
[GetSessionInformation operation](#) 65
[GetSheetNames operation](#) 67

[GetWorkbook operation](#) 70
[initialization](#) 35
[local events](#) 103
[message processing](#) 35
[NewWorkbook operation](#) 72
[OpenWorkbook operation](#) 75
[OpenWorkbookForEditing operation](#) 78
[overview](#) 34
[Refresh operation](#) 81
[SaveWorkbook operation](#) 82
[SaveWorkbookCopy operation](#) 84
[sequencing rules](#) 35
[SetCalculationOptions operation](#) 87
[SetCell operation](#) 90
[SetCellA1 operation](#) 92
[SetParameters operation](#) 94
[SetRange operation](#) 98
[SetRangeA1 operation](#) 100
[timer events](#) 103
Session timeout
[timers](#) 35
[SheetName simple type](#) 22
[Simple types](#) 21
[CellError](#) 21
[RangeReference](#) 22
[SheetName](#) 22
[WorkbookType](#) 23
[Standards assignments](#) 17
[Status complex type](#) 20
Syntax
[messages - overview](#) 18

T

Timer events
[server](#) 103
Timers
[request timeout](#) 35
[session timeout](#) 35
[Tracking changes](#) 148
[Transport](#) 18
[TypeExporter complex type](#) 21
Types
[complex](#) 19
[simple](#) 21

V

[Vendor-extensible fields](#) 17
[Versioning](#) 17

W

[WorkbookType simple type](#) 23
[WSDL](#) 120