

[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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplq@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
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.

Table of Contents

1	Introduction	9
1.1	Glossary	9
1.2	References.....	11
1.2.1	Normative References.....	11
1.2.2	Informative References	12
1.3	Protocol Overview (Synopsis)	12
1.3.1	Published Items and Results Returned by this Protocol	13
1.4	Relationship to Other Protocols.....	13
1.5	Prerequisites/Preconditions	13
1.6	Applicability Statement.....	13
1.7	Versioning and Capability Negotiation.....	13
1.8	Vendor-Extensible Fields.....	14
1.9	Standards Assignments	14
2	Messages.....	15
2.1	Transport.....	15
2.2	Common Message Syntax	15
2.2.1	Namespaces	15
2.2.2	Messages	15
2.2.3	Elements.....	16
2.2.4	Complex Types	16
2.2.4.1	ArrayOfAnyType	16
2.2.4.2	ArrayOfStatus	16
2.2.4.3	RangeCoordinates.....	17
2.2.4.4	Status	17
2.2.4.5	TypeExporter	18
2.2.5	Simple Types.....	18
2.2.5.1	CellError.....	18
2.2.5.2	RangeReference	19
2.2.5.3	SheetName.....	19
2.2.5.4	WorkbookType	19
2.2.6	Attributes.....	20
2.2.7	Groups.....	20
2.2.8	Attribute Groups	20
2.2.9	Faults	20
2.2.9.1	List of Subcodes	20
2.2.9.2	Common List of Subcodes.....	29
3	Protocol Details.....	31
3.1	Server Details	32
3.1.1	Abstract Data Model	32
3.1.2	Timers	32
3.1.2.1	Session Timeout.....	32
3.1.2.2	Request Timeout	32
3.1.3	Initialization	32
3.1.4	Message Processing Events and Sequencing Rules.....	32
3.1.4.1	Calculate	34
3.1.4.1.1	Messages	35
3.1.4.1.1.1	CalculateSoapIn	35
3.1.4.1.1.2	CalculateSoapOut	35

3.1.4.1.2	Elements.....	35
3.1.4.1.2.1	Calculate	35
3.1.4.1.2.2	CalculateResponse.....	35
3.1.4.2	CalculateA1.....	36
3.1.4.2.1	Messages	37
3.1.4.2.1.1	CalculateA1SoapIn.....	37
3.1.4.2.1.2	CalculateA1SoapOut	37
3.1.4.2.2	Elements.....	37
3.1.4.2.2.1	CalculateA1	37
3.1.4.2.2.2	CalculateA1Response	37
3.1.4.3	CalculateWorkbook	38
3.1.4.3.1	Messages	38
3.1.4.3.1.1	CalculateWorkbookSoapIn	38
3.1.4.3.1.2	CalculateWorkbookSoapOut	39
3.1.4.3.2	Elements.....	39
3.1.4.3.2.1	CalculateWorkbook	39
3.1.4.3.2.2	CalculateWorkbookResponse.....	39
3.1.4.3.3	Simple Types.....	39
3.1.4.3.3.1	CalculateType	39
3.1.4.4	CancelRequest	40
3.1.4.4.1	Messages	40
3.1.4.4.1.1	CancelRequestSoapIn	40
3.1.4.4.1.2	CancelRequestSoapOut	41
3.1.4.4.2	Elements.....	41
3.1.4.4.2.1	CancelRequest	41
3.1.4.4.2.2	CancelRequestResponse	41
3.1.4.5	CloseWorkbook	41
3.1.4.5.1	Messages	42
3.1.4.5.1.1	CloseWorkbookSoapIn	42
3.1.4.5.1.2	CloseWorkbookSoapOut	42
3.1.4.5.2	Elements.....	42
3.1.4.5.2.1	CloseWorkbook	42
3.1.4.5.2.2	CloseWorkbookResponse	42
3.1.4.6	GetApiVersion	43
3.1.4.6.1	Messages	43
3.1.4.6.1.1	GetApiVersionSoapIn	43
3.1.4.6.1.2	GetApiVersionSoapOut	43
3.1.4.6.2	Elements.....	44
3.1.4.6.2.1	GetApiVersion	44
3.1.4.6.2.2	GetApiVersionResponse	44
3.1.4.7	GetCell	44
3.1.4.7.1	Messages	45
3.1.4.7.1.1	GetCellSoapIn	45
3.1.4.7.1.2	GetCellSoapOut.....	45
3.1.4.7.2	Elements.....	45
3.1.4.7.2.1	GetCell	45
3.1.4.7.2.2	GetCellResponse.....	46
3.1.4.8	GetCellA1	46
3.1.4.8.1	Messages	47
3.1.4.8.1.1	GetCellA1SoapIn	47
3.1.4.8.1.2	GetCellA1SoapOut	47
3.1.4.8.2	Elements.....	48
3.1.4.8.2.1	GetCellA1	48

3.1.4.8.2.2	GetCellA1Response	48
3.1.4.9	GetChartImageUrl	48
3.1.4.9.1	Messages	49
3.1.4.9.1.1	GetChartImageUrlSoapIn	49
3.1.4.9.1.2	GetChartImageUrlSoapOut	49
3.1.4.9.2	Elements	49
3.1.4.9.2.1	GetChartImageUrl	49
3.1.4.9.2.2	GetChartImageUrlResponse	50
3.1.4.9.3	Complex Types	50
3.1.4.9.3.1	Size	50
3.1.4.10	GetPublishedItemNames	50
3.1.4.10.1	Messages	51
3.1.4.10.1.1	GetPublishedItemNamesSoapIn	51
3.1.4.10.1.2	GetPublishedItemNamesSoapOut	51
3.1.4.10.2	Elements	51
3.1.4.10.2.1	GetPublishedItemNames	51
3.1.4.10.2.2	GetPublishedItemNamesResponse	51
3.1.4.10.3	Complex Types	52
3.1.4.10.3.1	ArrayOfWorkbookItem	52
3.1.4.10.3.2	WorkbookItem	52
3.1.4.10.4	Simple Types	52
3.1.4.10.4.1	ItemType	52
3.1.4.11	GetRange	53
3.1.4.11.1	Messages	54
3.1.4.11.1.1	GetRangeSoapIn	54
3.1.4.11.1.2	GetRangeSoapOut	54
3.1.4.11.2	Elements	54
3.1.4.11.2.1	GetRange	54
3.1.4.11.2.2	GetRangeResponse	55
3.1.4.12	GetRangeA1	55
3.1.4.12.1	Messages	56
3.1.4.12.1.1	GetRangeA1SoapIn	56
3.1.4.12.1.2	GetRangeA1SoapOut	57
3.1.4.12.2	Elements	57
3.1.4.12.2.1	GetRangeA1	57
3.1.4.12.2.2	GetRangeA1Response	57
3.1.4.13	GetSessionInformation	58
3.1.4.13.1	Messages	58
3.1.4.13.1.1	GetSessionInformationSoapIn	58
3.1.4.13.1.2	GetSessionInformationSoapOut	58
3.1.4.13.2	Elements	58
3.1.4.13.2.1	GetSessionInformation	58
3.1.4.13.2.2	GetSessionInformationResponse	59
3.1.4.14	GetSheetNames	59
3.1.4.14.1	Messages	60
3.1.4.14.1.1	GetSheetNamesSoapIn	60
3.1.4.14.1.2	GetSheetNamesSoapOut	60
3.1.4.14.2	Elements	60
3.1.4.14.2.1	GetSheetNames	60
3.1.4.14.2.2	GetSheetNamesResponse	60
3.1.4.14.3	Complex Types	61
3.1.4.14.3.1	ArrayOfSheetInfo	61
3.1.4.14.3.2	SheetInfo	61

3.1.4.14.4	Simple Types	61
3.1.4.14.4.1	SheetType	61
3.1.4.14.4.2	SheetVisibility	62
3.1.4.15	GetWorkbook	62
3.1.4.15.1	Messages	63
3.1.4.15.1.1	GetWorkbookSoapIn	63
3.1.4.15.1.2	GetWorkbookSoapOut	63
3.1.4.15.2	Elements	63
3.1.4.15.2.1	GetWorkbook	63
3.1.4.15.2.2	GetWorkbookResponse	64
3.1.4.16	NewWorkbook	64
3.1.4.16.1	Messages	64
3.1.4.16.1.1	NewWorkbookSoapIn	64
3.1.4.16.1.2	NewWorkbookSoapOut	65
3.1.4.16.2	Elements	65
3.1.4.16.2.1	NewWorkbook	65
3.1.4.16.2.2	NewWorkbookResponse	65
3.1.4.17	OpenWorkbook	66
3.1.4.17.1	Messages	67
3.1.4.17.1.1	OpenWorkbookSoapIn	67
3.1.4.17.1.2	OpenWorkbookSoapOut	67
3.1.4.17.2	Elements	67
3.1.4.17.2.1	OpenWorkbook	67
3.1.4.17.2.2	OpenWorkbookResponse	68
3.1.4.18	OpenWorkbookForEditing	68
3.1.4.18.1	Messages	70
3.1.4.18.1.1	OpenWorkbookForEditingSoapIn	70
3.1.4.18.1.2	OpenWorkbookForEditingSoapOut	70
3.1.4.18.2	Elements	70
3.1.4.18.2.1	OpenWorkbookForEditing	70
3.1.4.18.2.2	OpenWorkbookForEditingResponse	70
3.1.4.19	Refresh	71
3.1.4.19.1	Messages	71
3.1.4.19.1.1	RefreshSoapIn	71
3.1.4.19.1.2	RefreshSoapOut	71
3.1.4.19.2	Elements	72
3.1.4.19.2.1	Refresh	72
3.1.4.19.2.2	RefreshResponse	72
3.1.4.20	SaveWorkbook	72
3.1.4.20.1	Messages	73
3.1.4.20.1.1	SaveWorkbookSoapIn	73
3.1.4.20.1.2	SaveWorkbookSoapOut	73
3.1.4.20.2	Elements	73
3.1.4.20.2.1	SaveWorkbook	73
3.1.4.20.2.2	SaveWorkbookResponse	74
3.1.4.21	SaveWorkbookCopy	74
3.1.4.21.1	Messages	75
3.1.4.21.1.1	SaveWorkbookCopySoapIn	75
3.1.4.21.1.2	SaveWorkbookCopySoapOut	75
3.1.4.21.2	Elements	75
3.1.4.21.2.1	SaveWorkbookCopy	75
3.1.4.21.2.2	SaveWorkbookCopyResponse	76
3.1.4.21.3	Simple Types	76

3.1.4.21.3.1	SaveOptions.....	76
3.1.4.22	SetCalculationOptions.....	77
3.1.4.22.1	Messages	77
3.1.4.22.1.1	SetCalculationOptionsSoapIn.....	77
3.1.4.22.1.2	SetCalculationOptionsSoapOut	77
3.1.4.22.2	Elements	77
3.1.4.22.2.1	SetCalculationOptions	77
3.1.4.22.2.2	SetCalculationOptionsResponse	78
3.1.4.22.3	Simple Types	78
3.1.4.22.3.1	WorkbookCalculation.....	78
3.1.4.23	SetCell	79
3.1.4.23.1	Messages	79
3.1.4.23.1.1	SetCellSoapIn	79
3.1.4.23.1.2	SetCellSoapOut	80
3.1.4.23.2	Elements	80
3.1.4.23.2.1	SetCell	80
3.1.4.23.2.2	SetCellResponse	80
3.1.4.24	SetCellA1.....	81
3.1.4.24.1	Messages	82
3.1.4.24.1.1	SetCellA1SoapIn.....	82
3.1.4.24.1.2	SetCellA1SoapOut.....	82
3.1.4.24.2	Elements	82
3.1.4.24.2.1	SetCellA1.....	82
3.1.4.24.2.2	SetCellA1Response	83
3.1.4.25	SetParameters.....	83
3.1.4.25.1	Messages	84
3.1.4.25.1.1	SetParametersSoapIn.....	84
3.1.4.25.1.2	SetParametersSoapOut	84
3.1.4.25.2	Elements	84
3.1.4.25.2.1	SetParameters	84
3.1.4.25.2.2	SetParametersResponse	85
3.1.4.25.3	Complex Types	85
3.1.4.25.3.1	ArrayOfParameterInfo	85
3.1.4.25.3.2	ParameterInfo	85
3.1.4.25.3.3	ArrayOfString.....	86
3.1.4.26	SetRange.....	86
3.1.4.26.1	Messages	87
3.1.4.26.1.1	SetRangeSoapIn.....	87
3.1.4.26.1.2	SetRangeSoapOut.....	87
3.1.4.26.2	Elements	87
3.1.4.26.2.1	SetRange.....	87
3.1.4.26.2.2	SetRangeResponse	88
3.1.4.27	SetRangeA1	88
3.1.4.27.1	Messages	89
3.1.4.27.1.1	SetRangeA1SoapIn	89
3.1.4.27.1.2	SetRangeA1SoapOut	89
3.1.4.27.2	Elements	89
3.1.4.27.2.1	SetRangeA1	89
3.1.4.27.2.2	SetRangeA1Response.....	90
3.1.5	Timer Events	90
3.1.6	Other Local Events	90

4 Protocol Examples..... 91

4.1	A Simple Example: Getting the Value of a Worksheet Cell.....	91
4.2	Advanced Example: Refreshing Data and Retrieving a Workbook	94
5	Security.....	106
5.1	Security Considerations for Implementers.....	106
5.2	Index of Security Parameters	106
6	Appendix A: Full WSDL	107
7	Appendix B: Product Behavior	133
8	Change Tracking.....	135
9	Index	136

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.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Transmission Control Protocol (TCP)
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
item
level
license key
list
locked
manual calculation mode
manual filter
merged cell
named object
named range
new workbook session
non-contiguous range
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

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

1.2.1 Normative References

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

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

[MS-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](#)"

[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

[MC-NMF] Microsoft Corporation, "[.NET Message Framing Protocol Specification](#)".

[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)

The Excel Web Services 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 isn't 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 [\[RFC2616\]](#), **HTTPS**, as described in [\[RFC2818\]](#), or **TCP**, as described in [\[MC-NMF\]](#).

The following diagram shows the underlying messaging and transport stack that the protocol uses:

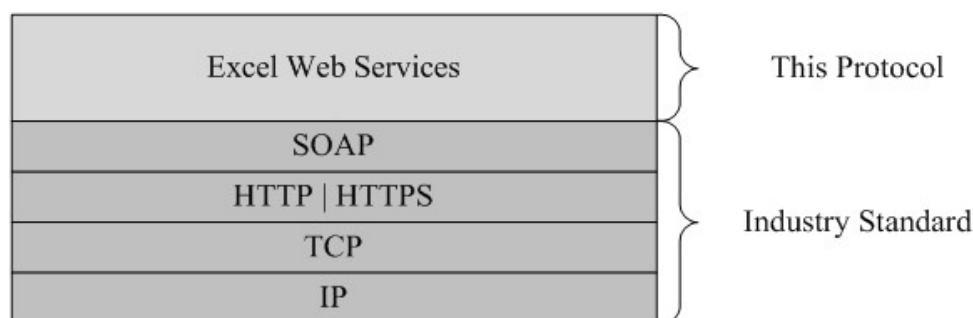


Figure 1: This protocol in relation to other protocols

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 anytime access to 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 be less restrictive than 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 specifies additional restrictions 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 structures used by this protocol. The syntax of the structures uses XML Schema, as specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language, 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

None.

2.2.3 Elements

None.

2.2.4 Complex Types

The following table summarizes the set of common XML Schema (1) complex type definitions defined by this specification. XML Schema (1) 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.

```
<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.

```
<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.

```
<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 1048575.

Column: The **column (2)** index. MUST be greater than or equal to 0 and less than or equal to 16383.

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 1048576.

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 16384.

2.2.4.4 Status

The status information returned by the protocol server.

```
<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 [List of Subcodes](#), but MAY be any other string.

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

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

Message: The message description.

2.2.4.5 TypeExporter

MUST NOT be used.

```
<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 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.

```
<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>
```

The following table specifies the allowable values for CellError:

Value	Meaning
Div0	Worksheet cell contains a number that is divided by 0.

Value	Meaning
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.

```
<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.

```
<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.

```

<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

None.

2.2.7 Groups

None.

2.2.8 Attribute Groups

None.

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

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 (Hypertext Transfer Protocol over Secure Sockets

Value	Meaning
	Layer), and the protocol client has tried to connect with a non HTTPS (Hypertext Transfer Protocol over Secure Sockets Layer).
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.
ExternalDataNotAllowedOnTL	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 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.

Value	Meaning
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.
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.

Value	Meaning
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.
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

Value	Meaning
	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 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

Value	Meaning
	workbook file.
IRMedWorkbook	An attempt to open a workbook failed because the workbook is in an Information Rights Management 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.
DimensionMismatch	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 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.

Value	Meaning
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	MUST NOT be used.
ApiInvalidCoordinate	MUST NOT be used.
ConfirmationUnsupported	MUST NOT be used.
ApiInvalidDimensionReturned	MUST NOT be used.
ApiMissingSiteContext	MUST NOT be used.
Ewr_IllegalPageSize	MUST NOT be used.
Ewr_WorkbookNotSpecified	MUST NOT be used.
Ewr_LoadProgress	MUST NOT be used.
Ewr_FileCorrupt	MUST NOT be used.
Ewr_WorkbookNotSupported	MUST NOT be used.
Ewr_GenericFileOpenError	MUST NOT be used.
Ewr_InvalidParameterGeneric	MUST NOT be used.
Ewr_InvalidParameterMandatoryMissing	MUST NOT be used.

Value	Meaning
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 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.

Value	Meaning
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.
MissingArgumentName	MUST NOT be used.
InvalidArgumentName	MUST NOT be used.
ArgumentBadInRemove	MUST NOT be used.
ManagementUICollectionEntryAlreadyExists	MUST NOT be used.
ManagementUICollectionEntryNotFound	MUST NOT be used.

Value	Meaning
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

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

3 Protocol Details

In the following sections, the schema definition might be less restrictive than 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 specifies additional restrictions 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.

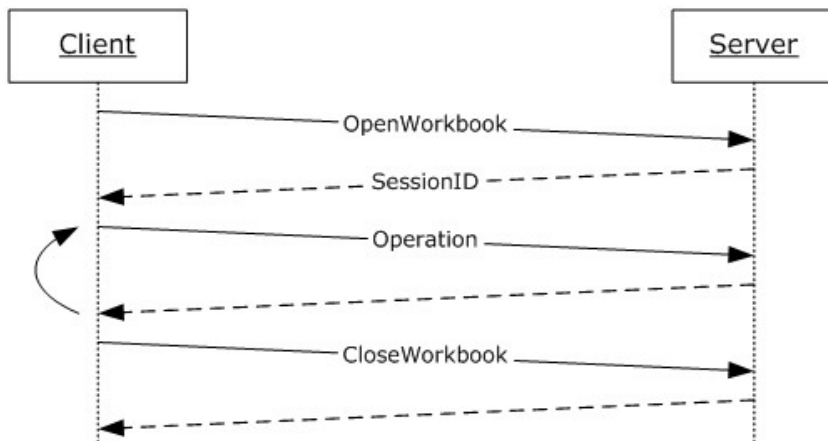


Figure 2: Excel Web Services Protocol high-level sequence diagram

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.

NewWorkbook 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](#). A new participant is added to the edit session on a successful **OpenWorkbookForEditing** request.

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

Request Behavior

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

3.1 Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

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

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

3.1.2 Timers

3.1.2.1 Session Timeout

The protocol server MAY [<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 is a list of interfaces belonging to Group 1:

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 is a list of interfaces belonging to Group 2:

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.

WSDL Operation	Description
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.

```
<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. **sheetName** MUST NOT be empty if **rangeCoordinates** isn't empty, as defined in [RangeCoordinates](#). If **sheetName** is empty, the whole workbook MUST be calculated.
2. If **rangeCoordinates** is empty, as defined in RangeCoordinates, and the **sheetName** is not empty, the whole sheet, which was specified in **sheetName**, MUST be calculated.
3. If **rangeCoordinates** is not empty, the specified range MUST be calculated.
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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
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](#).

3.1.4.2 CalculateA1

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

```

<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. **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.
2. **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.
3. If **rangeName** is not empty, the specified range MUST be calculated.
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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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 [RangeReference](#).

3.1.4.2.2.2 CalculateA1Response

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

```

<xs:element name="CalculateAlResponse">
  <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](#).

3.1.4.3 CalculateWorkbook

This operation is used to calculate the workbook.

```

<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 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.

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 [CalculateType](#).

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 [ArrayOfStatus](#).

3.1.4.3.3 Simple Types

3.1.4.3.3.1 CalculateType

```
<xs:simpleType name="CalculateType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Recalculate"/>
  </xs:restriction>
</xs:simpleType>
```

```

        <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.

3.1.4.4 CancelRequest

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

```

<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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/webservicesCancelRequest
```

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/webservicesCancelRequest
```

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](#).

3.1.4.5 CloseWorkbook

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

```
<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 Common List of Subcodes, but MAY instead contain any subcode specified in List of Subcodes 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.

3.1.4.6 GetApiVersion

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

```

<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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/webservicesGetApiVersion
```

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/webservicesGetApiVersion
```

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.

3.1.4.7 GetCell

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

```
<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](#).
3. If **formatted** is true, the sheet cell returned as the **GetCellResult** element MUST be type **string**, and MUST be number formatted 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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:

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

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>
```

```
</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 [SheetName](#) for specification. 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 1048575.

column: The column (2) index of the cell. MUST be greater than or equal to 0 and less than or equal to 16383

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 [ArrayOfStatus](#).

3.1.4.8 GetCellA1

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

```
<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 cells to retrieve. The sheet which contains the sheet cell which MUST be retrieved 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 [SheetName](#).

- 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](#).
- 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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>
```

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 [SheetName](#) for specification. MUST be present.

rangeName: The name of the range as specified in [RangeReference](#). 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 [ArrayOfStatus](#).

3.1.4.9 GetChartImageUrl

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

```
<wsdl:operation name="GetChartImageUrl">
  <wsdl:input message="tns:GetChartImageUrlSoapIn"/>
  <wsdl:output message="tns:GetChartImageUrlSoapOut"/>
</wsdl:operation>
```



```
</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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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/webservicesGetChartImageUrl
```

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/webservicesGetChartImageUrl
```

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 [Size](#). 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 [ArrayOfStatus](#).

3.1.4.9.3 Complex Types

3.1.4.9.3.1 Size

Specifies the size of a chart image.

```
<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.

3.1.4.10 GetPublishedItemNames

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

```
<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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 [ArrayOfWorkbookItem](#). MUST be present.

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

3.1.4.10.3 Complex Types

3.1.4.10.3.1 ArrayOfWorkbookItem

An array of **WorkbookItem** elements.

```

<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 [WorkbookItem](#).

3.1.4.10.3.2 WorkbookItem

Specifies the name and type of a published item.

```

<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 [ItemType](#).

3.1.4.10.4 Simple Types

3.1.4.10.4.1 ItemType

Specifies the type of a published item.

```

<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.

3.1.4.11 GetRange

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

```

<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** and 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](#).
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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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>
```

```
</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 [SheetName](#) for specification. MUST be present. MUST NOT be empty.

rangeCoordinates: The range to retrieve.

- MUST be present.
- MUST NOT be empty, as specified in [RangeCoordinates](#).
- The product of **Height** and **Width** MUST NOT be greater than 1000000.

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 [ArrayOfStatus](#).

3.1.4.12 GetRangeA1

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

```
<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 that MUST be retrieved 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 [SheetName](#). 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** and 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](#).
- 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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 [SheetName](#) for specification. MUST be present.

rangeName: The name of the range as specified in [RangeReference](#). 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>
```

GetRangeA1Result: The returned range value.

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

3.1.4.13 GetSessionInformation

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

```
<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 [ArrayOfStatus](#).

3.1.4.14 GetSheetNames

The operation returns information about the sheets in a workbook.

```

<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

3.1.4.14.1 Messages

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 [ArrayOfSheetInfo](#). MUST be present.

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

3.1.4.14.3 Complex Types

3.1.4.14.3.1 ArrayOfSheetInfo

An array of **SheetInfo** elements.

```

<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 [SheetInfo](#).

3.1.4.14.3.2 SheetInfo

Specifies the **name**, **type** and visible condition of a sheet.

```

<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 [SheetType](#).

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

3.1.4.14.4 Simple Types

3.1.4.14.4.1 SheetType

Specifies the type of a sheet.

```

<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.

```

<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.

3.1.4.15 GetWorkbook

This operation is used to retrieve a workbook file.

```

<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 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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>
```

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 [WorkbookType](#).

3.1.4.15.2.2 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 [ArrayOfStatus](#).

3.1.4.16 NewWorkbook

This operation is used to create a new workbook session.

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

1. 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.
2. 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.
3. 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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: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.

3.1.4.17 OpenWorkbook

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

```
<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 [\[MS-XLSB\]](#), [\[ISO/IEC-29500-1\]](#), or [\[MS-OFFMACRO2\]](#) 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

Subcode
FileOpenNotFound
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError

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: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.

3.1.4.17.2.2 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.

3.1.4.18 OpenWorkbookForEditing

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

```

<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 [\[MS-XLSB\]](#), [\[ISO/IEC-29500-1\]](#) or [\[MS-OFFMACRO2\]](#), 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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>
```

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 [ArrayOfStatus](#).

3.1.4.19 Refresh

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

```

<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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:

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>
```

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 [ArrayOfStatus](#).

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.

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


The protocol client sends a **SaveWorkbookSoapIn** request message and the protocol server responds with a **SaveWorkbookSoapOut** response message.

If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD 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.

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.

3.1.4.20.2.2 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 [ArrayOfStatus](#).

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.

```

<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.

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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

Subcode
EcsSnapshotAccessDenied
FileSaveGeneric
SaveFailed
SavingWorkbookAccessDenied
SavingWorkbookFileNameExists

Subcode
SavingWorkbookInvalidFileName
SavingWorkbookInvalidLocation
SavingWorkbookInvalidPermissions
SavingWorkbookLockedOrCheckedOut
SavingWorkbookNotCheckedOut
SavingWorkbookNotTrustedLocation
SavingWorkbookNoWorkbookName
SavingWorkbookTrustedLocationSettingWarning

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>
```

```

    </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 write to.

- MUST be present.

If the specified value is not a URI (Uniform Resource Identifier) 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 [WorkbookType](#).

saveOptions: A combination of 0 or more values of [SaveOptions](#). 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 [ArrayOfStatus](#).

3.1.4.21.3 Simple Types

3.1.4.21.3.1 SaveOptions

An enumeration of flags that can be specified for the **SaveWorkbookCopy** operation.

```

<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.

3.1.4.22 SetCalculationOptions

This operation is used to set the workbook calculation mode.

```
<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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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 [ArrayOfStatus](#).

3.1.4.22.3 Simple Types

3.1.4.22.3.1 WorkbookCalculation

The **WorkbookCalculation** type is an enumeration of a set of options that specify how the protocol server calculates the workbook.

```

<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 .

Value	Meaning
Semiautomatic	Specifies workbook is in automatic calculation mode, except for data tables.

3.1.4.23 SetCell

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

```
<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 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.
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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) 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:

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

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 [SheetName](#) for specification. 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 1048575.

column: The column (2) index of the worksheet cell. MUST be greater than or equal to 0 and less than or equal to 16383.

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 [ArrayOfStatus](#).

3.1.4.24 SetCellA1

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

```
<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 MUST 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 [SheetName](#).
 - 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 [RangeReference](#), 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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

Subcode
FormulaParseFailed
InvalidSheetName
DimensionMismatch
SheetRangeMismatch

Subcode
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 [SheetName](#) for specification. MUST be present. MUST NOT be empty.

rangeName: The range of the worksheet cell to set. See [RangeReference](#) for specification. MUST be present.

cellValue: The value to set in the worksheet cell.

3.1.4.24.2.2 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 [ArrayOfStatus](#).

3.1.4.25 SetParameters

This operation to sets values of workbook parameters in a workbook.

```
<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](#) 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 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.

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 to be set in the workbook, as specified in [ArrayOfParameterInfo](#). 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 [ArrayOfStatus](#).

3.1.4.25.3 Complex Types

3.1.4.25.3.1 ArrayOfParameterInfo

An array of **ParameterInfo** elements. 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 [ParameterInfo](#).

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.

```
<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 [ArrayOfString](#).

3.1.4.25.3.3 ArrayOfString

An array of string values.

```
<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.

3.1.4.26 SetRange

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

```
<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 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.
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 [Common List of Subcodes](#), but MAY instead contain any subcode specified in [List of Subcodes](#) or any other subcode.

Subcode
FormulaParseFailed

Subcode
InvalidSheetName
DimensionMismatch
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 [SheetName](#) for specification. MUST be present. MUST NOT be empty.

rangeCoordinates: The range to set. See [RangeReference](#) for specification. 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 [ArrayOfStatus](#).

3.1.4.27 SetRangeA1

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

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

1. The sheet that contains the range that MUST 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 [SheetName](#).
 - 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 [RangeReference](#), 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.

```
<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>
```

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 [SheetName](#) for specification. MUST be present.

rangeName: The name of the range to set. See [RangeReference](#) for specification. MUST be present.

rangeValues: The values to set in the range specified by **rangeName**. MUST be present.

3.1.4.27.2.2 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 [ArrayOfStatus](#).

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following subsections contain examples of this protocol's usage.

For an example of a system that implements this protocol, see [\[MSDN-WDCAEWS\]](#).

4.1 A Simple Example: 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.

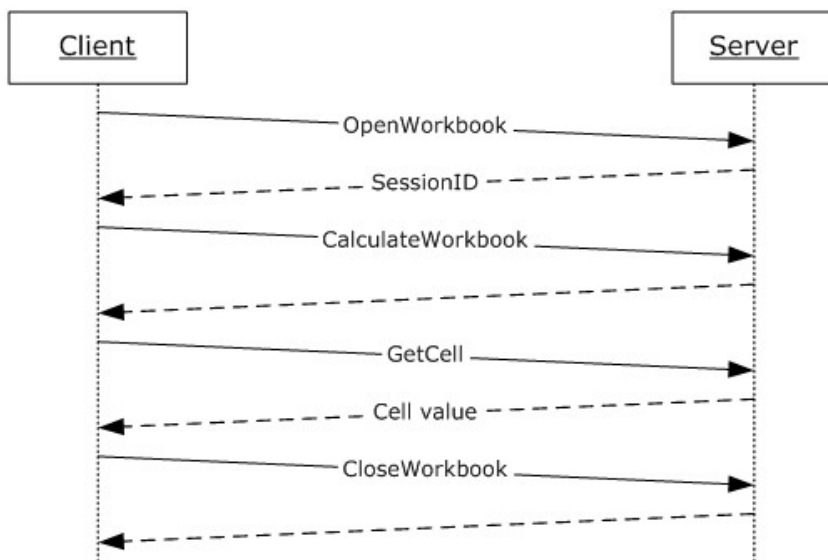


Figure 3: Get the value of a worksheet cell example

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.xls</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.V23.5791QXImNo/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 Advanced Example: 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.

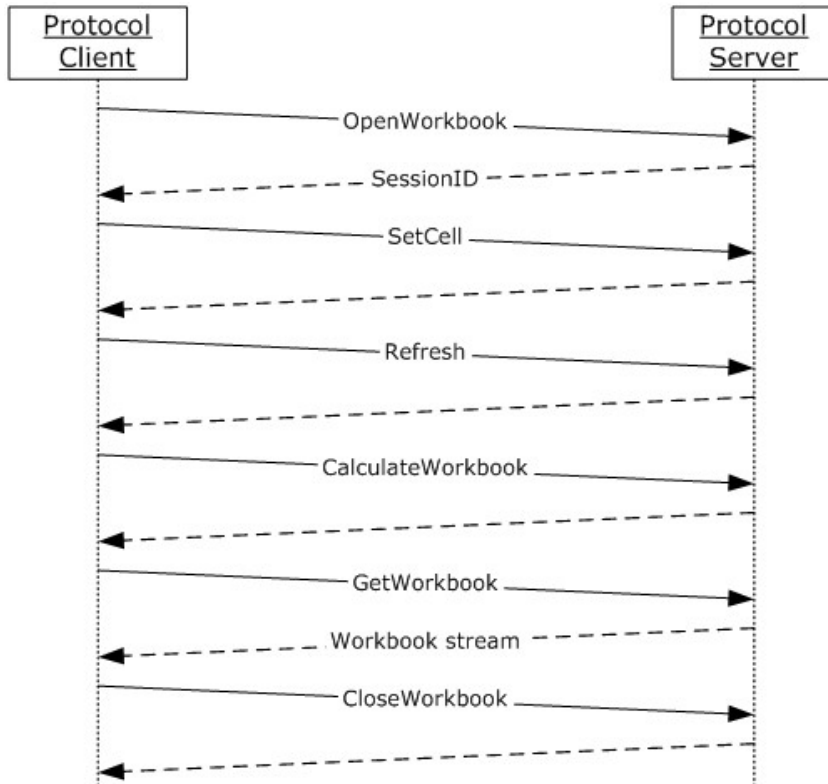


Figure 4: Refresh data and retrieve workbook example

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>

```

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"?>
```


98 / 137

MtI0gZbvSUBEot6fI/ve4/vfaJ87fqjIKFdIiTLccOrXC17iMQ+H9A4aHj3ep0rGx6SCscDzHhMGt6ESO/61ocfXMOBKl
QRQTA/lpu44YVKJZulkvRhGMurPCEX3BtyEWEfLyIoDQQ+Ar0RK62Uy2ulCNPYQzGOQO3d4ZD6BPW0Sm8rU95mcBkrqQd
8JrpaNXfMGnNbqKl15ES2mECmDU8sDPgRz3yShmIYangRsRmx+vtHWthDfTSUwtmJub1zE/6bx0wmC0YmyKoD81WulU
6+s7XgpJ5J8Fkwil0T154vMowYr2KaNqYuDxUORv3gxiLnCfAUKZWlYpzumNqC+45EN1FfSUuMEyQ1srWy/pX4N35rxZP
VPzTrTb7Va7kll7n5i1V499H3LERjEjfjWpno9JcimjkPLAf56PSKtFk1SVyxnhgnVmd2yf1ZrNZqy9DiuU8sB+rc85s1N
eq2yvL44zxwTm3Om2txutdaWxxnjgXVmbc6Zznpr9rbpEzhgPQkbj0ZwrunF10ssQl+n6h5ztFvqyAb5sAM2wXOF9bja
zF4AvTPmHDS6Qx2oRG4nwQy46IKAFGTT1GK1JQobYhy7ewlFfUKy5BN4kOHfHDvlybkjbQtIXNFEN76MEA2ea6Xvz8oc3
L5+jNy+fHT9+cFz45+MnT44f/2R1ORN3cRzkJ77+7vO/vvke/fn829dPvvyW13n533789NdfvigWBI41W9Grr579/uLZq
68/++P7pwXi20Bn8uI9GhGJ7pAjDMAJ8M0A466c9MX5ZvRCTJ0ZOATdBarbKnQE70wwK5JrEhe8+wLoZZHgjFDZ63dUI
wVLbB8K4wctW3OWROoxHEAT7StHMK9cRwUGxfjvNwBxodFtLs4dkLbHifAq7OkdLBvhcRZ5j7DscIBiYlCf4fEVLg3QN
KHVz3Mn6KH1DUxLQQkh7tO4k0m7RLI4jLpMhnCLWDzd591OSSyOsdguhKwobArGDxPcIcGG/gscJrKcoejlge8NtYhUWL
7E6En5drSWRWDgjjQd0gUhbNuSvA31zQb2EopoVh320TyJUUIo6KdN7GnOclD/ioFWJ48ChAoUvjMC97U44gRTHa56pIf
I+700rFQxxwvDDC9ylxwn16IbhHA2dJswTRd8ZCewGF2qm/EYUHQm7CYswoVGPr/WUxbnjb0JqKtsTuiRK8SG4JC+8OHS
f7BHJ9vvfC1t3Luu95+vuor18lmo7K7BQezV5sLZysYORuIYkHua6asLlBw14soRmMeJAOJ5J9FgtCeFjWtduUB
gMwcJrj6mKuyGOAGOXfG0kkCmqqOJEi7h9M8MF+rW8sDT1T07rOmnLVsPJFZ7fGCHV/WwGQdHp2pMywnMCWVmaFurOKux
1fVUKeh8G2MVvagzW6uYpZ1S51ibulzOGx00QQWgoC7AMprcLCoToCzCWZkoHG3DTGLi4nCRZYIhnhA0hhpv+djVDFBy
nLFHBdD7hTESD+EnoJazlpdq30Ha2cJUt5cdYG5LHrvEqUsg2dRAMontyOL85uTxeio4dVrKzUP+ThpeEN4rIWPuQJR15
r4YB8K2wCZv2p25ms8tn0axnjrmboAKHnhb30ZySroIuKzHnbb04gmFtpCrBYW7LrX6KBrBf1QsPbK4t5J9FgtCeFjWtduUB
RDS0ZDomv8sHOjWjs7GvASvLYEDENB0eoz8biAEP4daqCFWmq4XTCVAR9AS9bNNrmluc002XP6g3cqDjft4Q01BgloQ4
7QW6fmRlxvpiiswUIHOVvw6ALwTWIH9+nEO9+j/gnC8Alzhf3GvAk/18XpwlA4JTOnWBVuTDKlYbKs7DDY8LFXJoLkLI/
Y4APmhaAhQBeA8lt2Gf4tk81+QQ/3fllKrQ2tj8DCvDmiABEAwoUJByD50G1NUT1FWSSmJVZkpmDsx1tYZZG33ySfHPd
3aJ8t1A1fC7m0iB3G7mTO9e9TgtjP9DCNV9GtJ2Ys0rIuKzHnbb04gmFtpCrBYW7LrX6KBrBf1QsPbK4t5J9FgtCeFjWtduUB
ExnIdvp4WzLdcwjKZlG1EcX6v1LLFQRTnPYbBKc+Ft/Eh0n+Al1DhM/utBM2TevwAWiaCLxloZZA2kNVXLJ9Euu/ZwT7w
YTt0k0mrstCmJf1jlnGwC36Amdo9AbZe2VniF06wp5zbnfexsYsEO0XYwdqOLYQaIntyI8LQMHS+NyEXx2fJf+OE9x9Co
Hfgzc2YKWMScb5PIjA8UXTNPoDNby2aqVt/AwAA//8DAFBLAwQUAAACAAACEAs6QaM6YCAACYBgAAGAAAAHsL3dvcm
tzaG9lDHMvC2hl2XQxLnhtbIyVS2/iMBCA7yvtf7B8L014NAERqgQELsNKq+0+zsazEKtJnLUNTp9+zzYJDD+p8W8Aez3z
zsGeYp71VJtMA0kLWKY0GISVQc5mJepfS37/WDwk12rA6Y6WsIaXvoOnt4uuX+VGqV10AGIKewqe0MKaZBYHmBVRMD2QD
NZ7kUlXM4FbtAt0oYJkzqspgGlaPQcVETT1hpj7DkHkuOKwk31dQGw9RUDKD8etCNLq1VfWzuIqp133zwGXVIGIrSmHeH
ZSSis82uloqti0x77dozHjLdpsrfCW4klrmZoC4wAd6nXMSDAMkLeaZwAxs2YmCPKXP0WvHtRyZf2B/gg46g9rYtj2BU
rgBjK8J0ps/bdSv1rFDYpCRGgnYJGMG3GAJZRLStcTvMj/zgku0UHQefi4br2t3Y39UCSDnO1L81Mev4HYFQbOdOnMus9T
Fb1IJfDMjLBV7c79HkZKc1eLBJA5H0RADb0GbtC2lPC9NrL6e1KyKXSQ8QmCvy1k811I4ANySa2YYu5kkeCwld6obZ
txNxlgeboXPEa40i8k80GD2HD+o3ZlGmtcmSytN6ZASzEM787Azd9AVanRnUXfWQ2NqN9BW6mrXmQ87c4/2Gj7q0WR6H
27v+CrXpZX26aMtnfuf9GgS36E/3qRbaZ8+vgb7JRM9ic5V61Umvkm30j79fGe+M17jRI+TO3Q7wK4rY6V9+uNF7F7JR
9G58JL1Yp/epFTmcrRGRaoYkI8+nn8V5gH8qBMDZvnuvKteHk5s8dqDxx+itdgD2BmQ9f+w6FD2+Be3u215q6z+PqJ
d2sJGSW1LS4Nt3O4KHOMaFRUOsAdyKU27wfa23Bcw+4ZIJXBcuymd0kYqo5gw6GEmspSQTTZ0c6n7T1n8BwAA//8DAFBL
AwQUAAACAAACEALCPW3KgCAAA2BgAADQAAAHsL3N0eWxlcy54bWYkVN9r2zAQfh/sfXDK61l1bpY0wXZJMgKfRgzaw
V51W3bE9MNIckk69r/vJDUQx/a0ZfkdD59d/d9p4tvD1KgZ2Ys1yrB06sJRkzluuCuQsVDPp934BiPrqCqo0Iol+Mgsvk
0/f4qtOwr2ZuGfMCIYBQNZf75+oVITbFM0ntla6Zgi+1Np160JgUkZnNowllh/SQoSTSZiIlXueVYygf9tJXK4300l9rWvWsqeM
ZF9wdAxZGM1/dVUobmgko9TCd0fyEHQ6v4CXPjba6dFCAR3RZ8py9rvKGRASQ01glciedRblulAO2ehdq9vW4JzPMGqb
3uoCyhh9GY1GmKQx6a6nccanVJYp3pLF9Qc9UAMTUh+daaIMckAkgwaOoZG3ElgqeGe7DS1q50LbuyDsC/12c5MBGyN1mI
CEzXOJC9G1E0I23pDEQ6pHROzigzn46lpBegfYtTih7I7oy9DiNvg4ukJawJTntCpi1IYGtK40FKx1Ub3i19/901/Cbae
dAMQoK20ogFtMcrRGRaoYkI8+nn8V5gH8qBMDZvnuvKteHk5s8dqDxx+itdgD2BmQ9f+w6FD2+Be3u215q6z+PqJ
1LY4PjcyY2YWX1c3GBeq7Wj1j1vxZu00DVPsV4GX7CFqocC14pSQ7vRIY1/aI9trwF5DBz7mXPazJoexoB6IHal5o2auC
/DN8I8PvXsDz7phFWcOF46rn+awjYBaH82RMfE7nd0SYmT4LsFawkjBCPfUF3y2v7OCN3LZR/3wtAWIBJ/tez/A07nPw
Q7u3sK7hn/UGJ7gP9Fufb1YLtfj2WS5G8+iRT1e76434+3m23Y2ny7m19Hm72BlfWBhbcUKS5nKytgrZmu2a7F7MvW
NDW35QOcoR2UGfMCIYBQNZf75+oVITbFM0ntla6Zgi+1Np160JgUkZnNowllh/SQoSTSZiIlXueVYygf9tJXK4300l9rWvWsqeM
1ZXQZLnhtbIySTU/DMAyG70j8hyh3mo5vTesQEKLsgIT4umep20Zr4s4sjxGPx7vIwhpF242bHzzK/fzG4+w6g+gLLH20hJ
VwsF0WHRy9/ot9f7k2utMtvY2HejNPoLsr6ZHx/NNkirPACwEkLMjR6Y09SY7AYINleYIEqlQwGwJaXe5ERg23IpjOa0r
i9NsD7qHWFk/2Fg13kHd+jWASLVIASjZzk/Dz7lPS24/+CCpdU6nTgMSRBLP3r+K1Ctgpssu+ohkl6Po/pycW7dn1+QAH7
wjzNhxTizG/RQ87U5NUKazlovCrZrVwRdo28n2sxnZTnyHjb5T6y2u14irraFRdvoettqDnrvy66fSLXQ2fXIZ7h5AN8
PLMZe/NLvL5f5nmwPj5Z6H7MaoZurrSinn9JWZM5fRCqYUy9hng9gIYlddnWnVIFI++eGv+AK+TQvJiUXGm0QmJyXqW
F6ZeJNCiLYLN7z+afwMAAP//AwBQSwMEFAAGAAAGAAAHsL3N0eWxlcy54bWYkVN9r2zAQfh/sfXDK61l1bpY0wXZJMgKfRgzaw
2FjaGVEZWZpbml0aW9uMS54bWZcXVtz3Dayfj9V5z+w5mHf1rJkx052LW85kpWk1o69lpOtLZ99oGdoiZUZziyH41y760
+DIMBuoAECGGo48YutQV/Q+NBo3MFnf/uyWmafi3pbrqvz2emDh70sqObrRVndmM9+eX/1529n2bbJqW+XFF+eju27M7
+9vx//+fZpvy8bi7y+W1xWXwq7IBBRkoq7bns9um2fz15GQLxFW+fbDeFBVQPq3rVd7Az/rmZLupi3Qxv52KZrU8OXv4
8MnJKi+rmdTwlzPEx/rTp3JeXK7nu1VRNVJXsXzYcn2ttxswfL8c3GZN/n5DMpVF5/qArJcfH93Pnv5ZV4ss+ui/gw6g
FMTgR2K+ei7756cPnjy80Hjb0+/+e7p2e13s+xjPv/tp17vqsU/dkUNOK5n2RyKORSLLxWAj5Amnfh4lq3K6p3MPv+4LD
SLZz+v68UFqG1aK7e7zWZdn9e7j//RuXRpLxaf82peL7rcrK02U29/NhcVcL3e1fMia+42YHnxpSnqK1+CbeuqKuYcJZ8
WgvukY78qi+ViC+Q2y7NvlJo2PavYFwJ58LrItzuA698PPLzsts16VdRZa+S/QXG+EVrPZ5Qy6rd6mrViNwA7tuygPN6
fgtAnZ2ezbJl8bKAgx+dPX3ylNqi81TqcJ4/FOubOt/c3o1EYXF9RyyQSe6sod66jAVc29u8LhY/NcVK1/+JSM4+Q5FDs
3/wp3y1+euHFWbLnS/LXNhZPtM/RdmSFF7kVb7otMm/k1Vd1cJRPGHy72RVPxTQaqsW9PNZ9yN2S8QJ4pf9neILOv1st
PH0/ZVfYdlAe9wszTJJKD5B9Q+/2sbTej1uGaHe2R989t6/bnssFbNg1JcFPPM8s2qgXC4vS5W5evyS7F4D20bAis0K2j
VktTUBx12Bz9w6Jc1Altvfa7xOxH05r9drlt8iWepU9X6gph1BxSiMfTFuJgtlXmP0PQFT6briqATQwogAajhy7N5DQum8
1NsHwvSmhsFA3dyJAKP/ER69oQNZFauPRQONrtXQtlvZCU5Y+Lb1PPhgYIBqklJmGQXApLo60L4XgxHDMeka2pzwADae0
KILnhDvFoOnI6x43UKhuC8WCxgptZ2OasBdklnfonm4or85/n/3/X9bNbSYGzDh29alm6THlqwdgYr1a7Zoiuyy3MEWSAy
71ASbNBMOmfXWQCG/I1p+yq7LeNtlbmJjc51viHzyHCY+L66sA6eViN2+nqbjd6EQTCkT40koPM/1lxgTN15hgoUCJXWu

100 / 137

[MS-ESP] — v20110608
Excel Web Services Protocol Specification

[MS-ESP] — v20110608
Excel Web Services Protocol Specification
Copyright © 2011 Microsoft Corporation.
Release: Wednesday, June 8, 2011

ubImTDEA4hIc9RGcmf8xuErb3Uyf0W2spnt9NtuGqlunn7Bj9Rx07cZfi9DWpYhhI63Tajjgls4iZ8sHYGvBMKjVbLZfm
A0zo89jETN7eq7+vLMprNUrUWewGoYz9qoPCX39S3qn6H+LZRKYeyt5k2efubCBaw4QJHQII4o+MvCLf0giLezf1QFNiZ
oc1p+5kfmG11/MP8EAAD//wMAUESDBBQABgAIAAAAIQBeV9GgVQEAAGUGAAATACgAY3VzDG9tWG1sL210ZW0yLnhtbCCi
JAAoocAAAAAaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADs1U1PAjEQhv/KhjNud5FoJLm1Bq1SAxoh3rtlcbv7s
faADiE/leMqHHei8cJL00J100337TjVUCUD1zxY32qP1iU+OQgqtx8IYcxRqtS76V1G4Qx4p05X09M1tUChvZKcGmdWPu
VGEVcx17UR2Ncx2SuQ6HNNQmHGfNKx4UHFUDIZJ4+o0TjVlJbZDOYbzStrtPhhXhhNW98EB6R12iA0UmmeZvKzOZGvDHL8
Cao6zoEq021hcdzwI986hKuWGTvfax1YrWTFttU2Zzh/YqOwm7/H8UuqR99OsadlkFKQPFguNwVsmu8lrKKXgz7hZmE/U
xW2OdyXHP199v3fHgeX3hJfKzv1143QsQ2E1k42oyfKJ6aXeyGmXAMBMmpk1z0IGedGd28cmB+ss9HvnaYfC30hH1/QG
e9+/zT9WJgL/fPsvz1N//qf0ydHXwJ9BQAA//8DAFBLAWQUAAAYACAAACEAgGM8eBMJAA5KQAAEWaAoAGN1c3RvbVhtbC
9pdGVtMS54bWwgoiQAKKAgAAAAAaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA7Fpb9tGfN4v0P9AcJ816mZdjCi
FLTVY04b1GrRt2A4M7S4ITkMZxjLKP/75kbr6Is0Q76sgkQR9ScM+f6nQv97qdDHDnfamZDLqzd8XdkOjTBJITJ49rN
RTBYuj+9f4fFNWJoInYpaf0Ae9pjBx4+Hntuk6Mip+VQ7+imK7dLcN5DGTqVOXbu+3aHR1GY/g7ul1tZsv5ZjRd3I630
91s+fPmw+pm/mG2UzPvFjfbJu2fvlqQFe6ucN1SjrmWfUqTUaRoA5yEvkrECPHsEnygFkKcqrHxgxSNn+0WKLPZZDZe+
tPV0sc+wwf1L8Xc1X86u8HQCN4PdEn6Nxdrc5Feex5XVuhDOMQ24ywwQ8xiJvVBikK3GY3mXkWF1kggryKyZRSJZozSDKT
PREi5Yn4jRBB6uaDcfff/jD+8OnFxrqRyBskcqpE94ijAofLnQ5V3KWB1j0LvIcqo+BiGNCJemmxCOXE790aWL8RjN6GhF
8HxOCEKr6Rwt4HjCJzpita1BzEKep6en4dN0yLJHAbKx99fHex1t1k4Hfv7ZtDh6Gw+qamr5QNY1ixc4WKfGn1iMFuPB7
GoVDHx/gger1TwYX/1TTEd+YfQwTlkmnKQ091nknUajHMGsVg7VYksgcgstKIHmQy2kecfms0hc4vPdR42/j+iBD0q5o
Wwr3ihKLJsLWMBmtXOY5E+Dok9OBv4fJbxjnGfhk5LakP05YbXbl0Poke56LC27cocMv9Cfms0HkZd8LmJBS
k8P4m01/YCs098ulRQk/mozGt+n/D8uqZsbc0v/pcdev5RAqKrAu7IOCSmHSMQB9YFm9pgPIIsV5rjQIMh5s+p1TmMRl
vr+cx00o9wSkCLdipvg8RAiTGKVI7CUGLbPKBMJZTZQADMWlDdazvKzQbVT0CKJ+jE/LXjB3Ca1BKil+7maOeg6TAgr9r
N014HQYRciP4EqB8yTkaYSe68Xduds6f6LIVAJSCLcK7PXW+ya8CFjgCPLjNJQ3iPhxMKHEEG+9C7oSCXrPC10NCKDQkxc
UhdB9ZglJ9c0NpTktY9R6NueLwAcjRrQKkhlUCGxbBakSk4j82cgfzb+sd9pPoAPDxT6ogtoU1L3LuIBEAghJzY4AG2h4
AvRglshBEYfKlsJI14YWWssJ1GKSBR1JN80gGPghrQrGoolsFysW/OP3+6KeVGBRbSwlhUxKbXoIm5C0q0Iz9Doit9Lv
uY5mKm+UPk4eW8ZXTI6DaqFTgbtt3c9B37a+ZaWq1gRlmpoJF9KMOMIGdMokV8oTWUqgTcQIRdGRF21jrjgtWYQCh5Po
1vGIoqSV+RRR9EubTU+aiuBDixhsU1+ORLUwaKTr6VU7YEGlvKstnUV7nTvRBzTWUBVgatU33SHDezf2k2kwUwXG44FP6X
Qww4zpwJ/QxcCFzWky9BcYTzf/VTT23o9QTMKXnGsbK4j82cgfzb+sd9pPoAPDxT6ogtoU1L3LuIBEAghJzY4AG2h4
BkC4oUJ1higF4sliasdpGglePtqkcu+91F/Tarqidl+DoSPNxtjkaIzVyh6QqoBzFAMH2kRnw6I8TprtWpUrgtYVwmNB
J2hKKR6Mr+bbwQzN8AAtsrsggGC0XfkDGMCEz8GhFb01QvmdPKmDXrn2+hZ1S3f1E/Qsn1SNOayOfst4R1Ffxu9kzGHDvG
fuSp2fWANmrlFXAUfddnyc+y6ETabttBt8R0C9Ejy22y8jaPqKCAKEA0z5jiFuituAe3lvR2iVNP+PLzNg66bQpsQrmw
bdkpw1sjUGDInMMIAFMjshwdXWBL7YX2DChy2cmJqf7BFemuqQXSTVxr48gQx+fsTw12ii/BfMtWbKao1Y/aXSNVxCRTV
m61wmDcKEC6iEsIvSV5ByMEzzLFKWJNgzYd74+HYK8/CJFIZS6sE6pviJINoVRdm5PRqIV+67cqzn7ZmaYnkVfJga+dk
WBWk1bm+/J5hxaqQieQ+TbaYZVRpbET0wCbc+wr6gzRTbzTzRhOP4CGUGXLJc5YU1hxvcb3iVzehPoZJNT4V+1A7D2x2n
xtfFLNEpfazTVv7sB16znqm2keoJobgawXIKlh2so0p7rQig2K110bQrns1RqiLw3UIc7KQy5gJdJd20fTaIPOzJdqT
5brQ1p7Jru305vYN+JXd6LzNBzY+hedRGERzDqWxspUbp6KhsuGMWNC3in6TvY89/9Dsth5QSDxZ2fJGZf20fTaIPOzJdqT
OhJLHHLc+jyWABPSRZc8nac+QRWtmXpy8DbOMfvgvleuFCbkWmJgkTCubse7svsQ+dj78uYLSyyqIrlCaiKudVZLHPs3k
BoujB/CMZY4Vkg8dueJCaRpJArkVAsYwIacMdiwS03OgODt5Cq99YCWht1z6ChRA8jsU4X3BbbPjJd8e0yNQUwv9FNW01
W2JfnlgtbeIoz0VIS4+wts9WAty+2rv11AsNmXp+q0VzKVYn2BpI15AwUpEgkXpLoxpWZLqmhpkE4BYyHulnDLms+t6Xd
m0rV/1ru305vYN+JXd6LzNBzY+hedRGERzDqWxspUbp6KhsuGMWNC3in6TvY89/9Dsth5QSDxZ2fJGZf20fTaIPOzJdqT
l+3XSBbSmrFWLWIsa4ue6ugCRnysthxROPjLSNWTvAVHTU4/Lkqleb7YuLJmZdqaiag9iRen15LfbzQ3nDIEQpuRn6FDE
sw2FhhGkRqfGdDbwMh+6+oNwXtOzSdARcWMw3gL3wXhNWUfGCX+0bW714Ks8Z95yi05vk4ZnDe447M4L1K88mtSaFoTNfD
1aYlXroGtqX+BSPdgKspY5mmxs+90M15c4HLFNTlmapjQjTjQULrVAZeUbj2NXW7aWyLXethWt1Syux6YNmN9pQDP5p
8H9G1p7ojCM50mZr24pCGxIe9PCbqW3rWz2+8p89RjTtEYpOyqJ6rBx90WwaRxsKHI6t6spK/7E4Oz+xxDt/sTg7v7E40
/LyfeqbcRfRNdUt/B7w00Iu34TruV9W8Qf4UAR9FAVZXL80zdTRqqfy8ltYz9/Ct/h+hFSuWMYh/rHfvlx/f/AwAA//8D
AFBLAWQUAAAYACAAACEABW/ErBMAAADJAAAGAAoAGN1c3RvbVhtbC9pdGVtUHJvcHM0LnhtbCCiJAAoocAAAAAaAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjcsKwjaQRfeC/xBmX5MW1ViMorYB9wpuQzrVQjMjTRRB/Hezui+4Z7v/hF
G8cYoDk4FyoUAgee4GuHu4XmyhQcTKqHMjExoghv1Uptt2se5ccjHxhOeQEriYHpuDHzy61LW634yTloVejqpLW
rtLV2tWkP6gci0ynfRAOPlJ61lNE/MLi44CdSHnuegks5TnfJfT94bNi/AlKS1Vjr6V8ZH25hBlN7AwAA//8DAFBLAWQU
AAAYACAAACEA2jZYA7YAAADJAAAGAAoAGN1c3RvbVhtbC9pdGVtUHJvcHMzLnhtbCCiJAAoocAAAAAaAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjU0LwjaQRO+C/yHsPSbGrypGABEF7wpeQ7rVQrMrTRRB/O/mNMw8mLc/fsIg3jjGns
nCfKZBIHlue7pbuF4aWYCIyVhRbia0QAZhW3Syb+OudcnFxCoeEwaRhZ7n+WThW5i63mqwI/ViWcjlqmrklphaVtUc760ro
sy8asfyCymvNntPBI6blTKvoHBhdn/ETKsOmxuJTreFcdB3HE/tXQERKaL1W/pX14RYGUIC/AAAA//8DAFBLAWQUAAAY
CAAAACEA2UvLGLGAADJAAAGAAoAGN1c3RvbVhtbC9pdGVtUHJvcHMzLnhtbCCiJAAoocAAAAAaAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjcgKwjAURfcd/kN4+5ikjQWiqVjbgvsZmG1IX22heU+aKIL475PV5d4D9xxPr7CIJ65xZr
JgthoEkudhppuF359eViBicjS4hQktEMOp3nwdh3gYXHX8fYrXhEHKyc55bS28z13Zn1tTyEtXVvJ7X+xxD05pOm95qTNP
v9L7UHxhBZTfkmWphSuh+Uin7C4OKW70gzJrWGL3Jdb4rHcfbY8n8EpKQKrUv1H1kf/sICqv4HAAD//wMAUESDBBQABgAI
AAAAIQA3E8qItwAAAMKAAAYACgAY3VzdG9tWG1sL210ZWM1Qcm9wcZueG1sIKIkACigIAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAYNwYrCMBRF98L8Q3j7mCZTayTgcagF9wpuQ/qgheY9aeIgiP9uVpd7D9yz3b/CJP5xjiOTBb
0sQCB57ke6Wbic0lMdiM1R7yYmtEAM+93PytvHte+Si41nPCUMiG9jz1Nr4W3KqtDdYSXNnz7K81c3sq5Nkbt23Rh9rHV
V6g+IrKZ8EY3cU3p1Ir+jshFJT+QMhx4Di710t8UD8PocSWX/DehJamaKolH9mfbiGCdTUcWAA//8DAFBLAWQUAAAYACAA
ACEAksWzOhgBAAAKAgAAEWaAoAGN1c3RvbVhtbC9pdGVtNC54bWwgoiQAKKAgAAAAAaAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAARJHbSsNAEIZfpeX9sk1PMSFJkYIIWhDshbebyWyzkd2wOzX17U1rrYJFE1WbBr5v/pkpVwfdT17RB2VNxd
I4YRM0YftldhXbk4xu2KouXEG8dehJYZiMhAmFqlh8YArOa3SOrYilAm+DLRSD1dxKqQD5NEMWXCJOJVPdGxxZ211hyCuoi
GYyiHWWZ970il/GXz+HxyR8oEEgbwk3JwgX8droy0T1B39GX8Sxyg6NfWkLdY9HXZwthrNLQRuzuzWNXlVhzWgqC7fuf
jBWDGQu5DzKkiyN5otcRk0zhSjPlzJdNDPAPGH8hD7g22B900rut3cK+/ZPj1GBXp9P/D8r8pOxLvY6Ue4sX3tBPz7v
+t3AAAA//8DAFBLAQICtABQABgAIAAAAIQAEZCq/QEAAFYKAAATAAAAAAAAAAAAAAAAAAAAAABbQ29udGVudF9UeXB1c1
0ueG1sUESBAi0AFAAGAAgAAAAhABNevmUFAQAA3wIAAAsAAAAAAAAAAAAAAAAAAAJgQAAf9yZWxzLy5yZWxzUESBAi0AFAA

- Close the workbook session on the protocol server.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
    </CloseWorkbook>
  </soap:Body>
</soap:Envelope>
```

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

There are security risks associated with exposing session identifiers (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.

6 Appendix A: Full WSDL

```
<?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: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: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>
  </xs:element>
  <xs:complexType name="ArrayOfAnyType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="anyType" nillable="true" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="TypeExporter">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="CellError" type="tns:CellError" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="CellError">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Div0" />
      <xs:enumeration value="NA" />
      <xs:enumeration value="Name" />
      <xs:enumeration value="Null" />
      <xs:enumeration value="Num" />
      <xs:enumeration value="Ref" />
      <xs:enumeration value="Value" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="GetRangeA1">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetRangeA1Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="GetRangeA1Result"
type="tns:ArrayOfAnyType" />
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetCell">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int" />
        <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int" />
        <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetCellResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="GetCellResult" />
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="GetCellA1">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetCellA1Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="GetCellA1Result" />
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetPublishedItemNames">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="WorkbookItem">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ItemType" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ItemType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Chart" />
      <xs:enumeration value="PivotTable" />
      <xs:enumeration value="Table" />
      <xs:enumeration value="NamedRange" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ArrayOfWorkbookItem">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="WorkbookItem" nillable="true"
type="tns:WorkbookItem" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="GetPublishedItemNamesResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="GetPublishedItemNamesResult"
type="tns:ArrayOfWorkbookItem" />
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="GetSheetNames">
    <xs:complexType>
      <xs:sequence>

```



```

        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="SheetInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:SheetType" />
        <xs:element minOccurs="1" maxOccurs="1" name="Visibility"
type="tns:SheetVisibility" />
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="SheetType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Worksheet" />
        <xs:enumeration value="Chartsheet" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SheetVisibility">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Visible" />
        <xs:enumeration value="Hidden" />
        <xs:enumeration value="VeryHidden" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ArrayOfSheetInfo">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="SheetInfo" nillable="true"
type="tns:SheetInfo" />
    </xs:sequence>
</xs:complexType>
<xs:element name="GetSheetNamesResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetSheetNamesResult"
type="tns:ArrayOfSheetInfo" />
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetRange">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
type="tns:RangeCoordinates" />
            <xs:element minOccurs="0" maxOccurs="1" name="rangeValues"
type="tns:ArrayOfAnyType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="SetRangeResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

<xs:element name="SetRangeA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="rangeValues"
type="tns:ArrayOfAnyType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetRangeA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetCell">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int" />
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetCellResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetCellA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="SetCellA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element 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:sequence>
    </xs:complexType>
</xs:element>
<xs:simpleType name="CalculateType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Recalculate" />
        <xs:enumeration value="CalculateFull" />
    </xs:restriction>
</xs:simpleType>
<xs:element name="CalculateWorkbookResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetSessionInformation">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetSessionInformationResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="serverVersion" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureNameUsed" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureNameUsed"
type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CancelRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="CancelRequestResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetWorkbook">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
            <xs:element minOccurs="1" maxOccurs="1" name="workbookType"
type="tns:WorkbookType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="GetWorkbookResponse">

```

```

    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookResult"
type="xs:base64Binary" />
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetParameters">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfParameterInfo" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ArrayOfParameterInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ParameterInfo"
nillable="true" type="tns:ParameterInfo" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ParameterInfo">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Values" type="tns:ArrayOfString" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SetParametersResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="SetCalculationOptions">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
        <xs:element minOccurs="1" maxOccurs="1" name="calcType"
type="tns:WorkbookCalculation" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="WorkbookCalculation">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Manual" />
      <xs:enumeration value="Automatic" />
      <xs:enumeration value="Semiautomatic" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="SetCalculationOptionsResponse">

```

```

        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    <xs:element name="GetChartImageUrl">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="size" type="tns:Size" />
          <xs:element minOccurs="0" maxOccurs="1" name="chartName" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs: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>
<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>
        <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: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/OpenWorkbookForEditi
ng" style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook"
style="document" />
        <wsdl:input>
            <soap12:body use="literal" />
        </wsdl:input>
        <wsdl: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

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 [List of Subcodes](#).

<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.

9 Index

A

Abstract data model
 [server](#) 32
 [Applicability](#) 13
 [ArrayOfAnyType complex type](#) 16
 [ArrayOfStatus complex type](#) 16
 [Attribute groups](#) 20
 [Attributes](#) 20

C

[Capability negotiation](#) 13
[CellError simple type](#) 18
[Change tracking](#) 135
Client
 [overview](#) 31
 [Common list of subcodes - fault](#) 29
 [Complex types](#) 16
 [ArrayOfAnyType](#) 16
 [ArrayOfStatus](#) 16
 [RangeCoordinates](#) 17
 [Status](#) 17
 [TypeExporter](#) 18

D

Data model - abstract
 [server](#) 32

E

Events
 [local - server](#) 90
 [timer - server](#) 90
Examples
 [getting the value of a worksheet cell](#) 91
 [overview](#) 91
 [refreshing data and retrieving a workbook](#) 94

F

[Faults](#) 20
 [common list of subcodes](#) 29
 [list of subcodes](#) 20
[Fields - vendor-extensible](#) 14
[Full WSDL](#) 107

G

[Getting the value of a worksheet cell example](#) 91
[Glossary](#) 9
[Groups](#) 20

I

[Implementer - security considerations](#) 106
[Index of security parameters](#) 106
[Informative references](#) 12

Initialization
 [server](#) 32
 [Introduction](#) 9

L

[List of subcodes - fault](#) 20
Local events
 [server](#) 90

M

Message processing
 [server](#) 32
Messages
 [ArrayOfAnyType complex type](#) 16
 [ArrayOfStatus complex type](#) 16
 [attribute groups](#) 20
 [attributes](#) 20
 [CellError simple type](#) 18
 [common list of subcodes - fault](#) 29
 [complex types](#) 16
 [elements](#) 16
 [enumerated](#) 15
 [faults](#) 20
 [groups](#) 20
 [list of subcodes - fault](#) 20
 [namespaces](#) 15
 [RangeCoordinates complex type](#) 17
 [RangeReference simple type](#) 19
 [SheetName simple type](#) 19
 [simple types](#) 18
 [Status complex type](#) 17
 [syntax](#) 15
 [transport](#) 15
 [TypeExporter complex type](#) 18
 [WorkbookType simple type](#) 19

N

[Namespaces](#) 15
[Normative references](#) 11

O

Operations
 [Calculate](#) 34
 [CalculateA1](#) 36
 [CalculateWorkbook](#) 38
 [CancelRequest](#) 40
 [CloseWorkbook](#) 41
 [GetApiVersion](#) 43
 [GetCell](#) 44
 [GetCellA1](#) 46
 [GetChartImageUrl](#) 48
 [GetPublishedItemNames](#) 50
 [GetRange](#) 53
 [GetRangeA1](#) 55
 [GetSessionInformation](#) 58

- [GetSheetNames](#) 59
- [GetWorkbook](#) 62
- [NewWorkbook](#) 64
- [OpenWorkbook](#) 66
- [OpenWorkbookForEditing](#) 68
- [Refresh](#) 71
- [SaveWorkbook](#) 72
- [SaveWorkbookCopy](#) 74
- [SetCalculationOptions](#) 77
- [SetCell](#) 79
- [SetCellA1](#) 81
- [SetParameters](#) 83
- [SetRange](#) 86
- [SetRangeA1](#) 88
- [Overview \(synopsis\)](#) 12
 - [published items returned](#) 13
 - [results returned](#) 13

P

- [Parameters - security index](#) 106
- [Preconditions](#) 13
- [Prerequisites](#) 13
- [Product behavior](#) 133

R

- [RangeCoordinates complex type](#) 17
- [RangeReference simple type](#) 19
- References
 - [informative](#) 12
 - [normative](#) 11
- [Refreshing data and retrieving a workbook example](#) 94
- [Relationship to other protocols](#) 13
- Request timeout
 - [timers](#) 32
- [Results returned](#) 13

S

- Security
 - [implementer considerations](#) 106
 - [parameter index](#) 106
- Sequencing rules
 - [server](#) 32
- Server
 - [abstract data model](#) 32
 - [Calculate operation](#) 34
 - [CalculateA1 operation](#) 36
 - [CalculateWorkbook operation](#) 38
 - [CancelRequest operation](#) 40
 - [CloseWorkbook operation](#) 41
 - [GetApiVersion operation](#) 43
 - [GetCell operation](#) 44
 - [GetCellA1 operation](#) 46
 - [GetChartImageUrl operation](#) 48
 - [GetPublishedItemNames operation](#) 50
 - [GetRange operation](#) 53
 - [GetRangeA1 operation](#) 55
 - [GetSessionInformation operation](#) 58
 - [GetSheetNames operation](#) 59

- [GetWorkbook operation](#) 62
- [initialization](#) 32
- [local events](#) 90
- [message processing](#) 32
- [NewWorkbook operation](#) 64
- [OpenWorkbook operation](#) 66
- [OpenWorkbookForEditing operation](#) 68
- [overview](#) 31
- [Refresh operation](#) 71
- [SaveWorkbook operation](#) 72
- [SaveWorkbookCopy operation](#) 74
- [sequencing rules](#) 32
- [SetCalculationOptions operation](#) 77
- [SetCell operation](#) 79
- [SetCellA1 operation](#) 81
- [SetParameters operation](#) 83
- [SetRange operation](#) 86
- [SetRangeA1 operation](#) 88
- [timer events](#) 90
- Session timeout
 - [timers](#) 32
- [SheetName simple type](#) 19
- [Simple types](#) 18
 - [CellError](#) 18
 - [RangeReference](#) 19
 - [SheetName](#) 19
 - [WorkbookType](#) 19
- [Standards assignments](#) 14
- [Status complex type](#) 17
- Syntax
 - [messages - overview](#) 15

T

- Timer events
 - [server](#) 90
- Timers
 - [request timeout](#) 32
 - [session timeout](#) 32
- [Tracking changes](#) 135
- [Transport](#) 15
- [TypeExporter complex type](#) 18
- Types
 - [complex](#) 16
 - [simple](#) 18

V

- [Vendor-extensible fields](#) 14
- [Versioning](#) 13

W

- [WorkbookType simple type](#) 19
- [WSDL](#) 107