Excel Web Services Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

**Support.** For questions and support, please contact dochelp@microsoft.com.
## Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Revised and edited technical content</td>
</tr>
<tr>
<td>10/6/2008</td>
<td>1.01</td>
<td>Minor</td>
<td>Revised and edited technical content</td>
</tr>
<tr>
<td>12/12/2008</td>
<td>1.02</td>
<td>Minor</td>
<td>Revised and edited technical content</td>
</tr>
<tr>
<td>7/13/2009</td>
<td>1.03</td>
<td>Major</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>8/28/2009</td>
<td>1.04</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>11/6/2009</td>
<td>1.05</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>2/19/2010</td>
<td>2.0</td>
<td>Minor</td>
<td>Updated the technical content</td>
</tr>
<tr>
<td>3/31/2010</td>
<td>2.01</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>4/30/2010</td>
<td>2.02</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/7/2010</td>
<td>2.03</td>
<td>Editorial</td>
<td>Revised and edited the technical content</td>
</tr>
<tr>
<td>6/29/2010</td>
<td>2.04</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>7/23/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/27/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>11/15/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>12/17/2010</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/10/2011</td>
<td>2.04</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/11/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/12/2012</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>3.0.1</td>
<td>Editorial</td>
<td>Changed language and formatting in the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>3.0.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/30/2013</td>
<td>3.0.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>3.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>3.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>3.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>3.3</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>2/26/2016</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/15/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/20/2017</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/19/2019</td>
<td>7.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>6/18/2019</td>
<td>7.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
</tbody>
</table>
Table of Contents

1 Introduction .............................................................................................................. 13
  1.1 Glossary ................................................................................................................ 13
  1.2 References ............................................................................................................. 17
    1.2.1 Normative References .................................................................................. 17
    1.2.2 Informative References ............................................................................... 19
  1.3 Overview ................................................................................................................ 19
    1.3.1 Published Items and Results Returned by this Protocol ......................... 20
  1.4 Relationship to Other Protocols ........................................................................... 20
  1.5 Prerequisites/Preconditions .................................................................................. 20
  1.6 Applicability Statement ....................................................................................... 20
  1.7 Versioning and Capability Negotiation ............................................................... 20
  1.8 Vendor-Extensible Fields ..................................................................................... 21
  1.9 Standards Assignments ....................................................................................... 21

2 Messages ................................................................................................................ 22
  2.1 Transport ............................................................................................................... 22
  2.2 Common Message Syntax .................................................................................... 22
    2.2.1 Namespaces .................................................................................................... 22
    2.2.2 Messages ........................................................................................................ 22
    2.2.3 Elements ......................................................................................................... 23
    2.2.4 Complex Types .............................................................................................. 23
      2.2.4.1 ArrayOfAnyType .................................................................................... 23
      2.2.4.2 ArrayOfStatus ...................................................................................... 23
      2.2.4.3 ArrayOfString ....................................................................................... 24
      2.2.4.4 RangeCoordinates ............................................................................... 24
      2.2.4.5 Status ..................................................................................................... 24
      2.2.4.6 TypeExporter ......................................................................................... 25
    2.2.5 Simple Types .................................................................................................. 25
      2.2.5.1 CellError ................................................................................................ 26
      2.2.5.2 RangeReference ................................................................................... 26
      2.2.5.3 SaveOptions .......................................................................................... 27
      2.2.5.4 SheetName ............................................................................................ 27
      2.2.5.5 WorkbookType ..................................................................................... 28
    2.2.6 Attributes ........................................................................................................ 28
    2.2.7 Groups ............................................................................................................ 28
    2.2.8 Attribute Groups ............................................................................................ 28
    2.2.9 Common Data Structures .............................................................................. 29
      2.2.9.1 Faults ..................................................................................................... 29
      2.2.9.2 Common List of Subcodes ................................................................... 38

3 Protocol Details ..................................................................................................... 41
  3.1 Server Details ....................................................................................................... 42
    3.1.1 Abstract Data Model .................................................................................... 42
    3.1.2 Timers ............................................................................................................ 42
      3.1.2.1 Session Timeout ..................................................................................... 42
      3.1.2.2 Request Timeout ................................................................................... 42
    3.1.3 Initialization ................................................................................................... 42
    3.1.4 Message Processing Events and Sequencing Rules .................................... 42
      3.1.4.1 Calculate ............................................................................................... 44
        3.1.4.1.1 Messages ......................................................................................... 45
        3.1.4.1.1.1 CalculateSoapIn .......................................................................... 45
        3.1.4.1.1.2 CalculateSoapOut ........................................................................ 45
        3.1.4.1.2 Elements ......................................................................................... 45
        3.1.4.1.2.1 Calculate ....................................................................................... 45
        3.1.4.1.2.2 CalculateResponse ...................................................................... 46
3.1.4.1.3 Complex Types .......................................................... 46
3.1.4.1.4 Simple Types .......................................................... 46
3.1.4.1.5 Attributes .............................................................. 46
3.1.4.1.6 Groups .................................................................. 46
3.1.4.1.7 Attribute Groups ...................................................... 46
3.1.4.2 CalculateA1 ................................................................. 46
  3.1.4.2.1 Messages .............................................................. 47
    3.1.4.2.1.1 CalculateA1SoapIn ........................................... 48
    3.1.4.2.1.2 CalculateA1SoapOut ....................................... 48
  3.1.4.2.2 Elements .............................................................. 48
    3.1.4.2.2.1 CalculateA1 .................................................... 48
    3.1.4.2.2.2 CalculateA1Response ............................. 48
  3.1.4.2.3 Complex Types ...................................................... 49
  3.1.4.2.4 Simple Types ....................................................... 49
  3.1.4.2.5 Attributes ............................................................ 49
  3.1.4.2.6 Groups ................................................................ 49
  3.1.4.2.7 Attribute Groups ................................................. 49
3.1.4.3 CalculateWorkbook .................................................. 49
  3.1.4.3.1 Messages ............................................................ 50
    3.1.4.3.1.1 CalculateWorkbookSoapIn ............................... 50
    3.1.4.3.1.2 CalculateWorkbookSoapOut ............................ 50
  3.1.4.3.2 Elements .............................................................. 50
    3.1.4.3.2.1 CalculateWorkbook ....................................... 51
    3.1.4.3.2.2 CalculateWorkbookResponse ........................ 51
  3.1.4.3.3 Complex Types ...................................................... 51
  3.1.4.3.4 Simple Types ....................................................... 51
    3.1.4.3.4.1 CalculateType ............................................... 51
  3.1.4.3.5 Attributes ............................................................ 52
  3.1.4.3.6 Groups ................................................................ 52
  3.1.4.3.7 Attribute Groups ................................................. 52
3.1.4.4 CancelRequest .......................................................... 52
  3.1.4.4.1 Messages ............................................................ 52
    3.1.4.4.1.1 CancelRequestSoapIn ..................................... 53
    3.1.4.4.1.2 CancelRequestSoapOut .................................. 53
  3.1.4.4.2 Elements .............................................................. 53
    3.1.4.4.2.1 CancelRequest ............................................... 53
    3.1.4.4.2.2 CancelRequestResponse ............................... 54
  3.1.4.4.3 Complex Types ...................................................... 54
  3.1.4.4.4 Simple Types ....................................................... 54
  3.1.4.4.5 Attributes ............................................................ 54
  3.1.4.4.6 Groups ................................................................ 54
  3.1.4.4.7 Attribute Groups ................................................. 54
3.1.4.5 CloseWorkbook ......................................................... 54
  3.1.4.5.1 Messages ............................................................ 54
    3.1.4.5.1.1 CloseWorkbookSoapIn ..................................... 55
    3.1.4.5.1.2 CloseWorkbookSoapOut .................................. 55
  3.1.4.5.2 Elements .............................................................. 55
    3.1.4.5.2.1 CloseWorkbook ............................................... 55
    3.1.4.5.2.2 CloseWorkbookResponse ............................... 56
  3.1.4.5.3 Complex Types ...................................................... 56
  3.1.4.5.4 Simple Types ....................................................... 56
  3.1.4.5.5 Attributes ............................................................ 56
  3.1.4.5.6 Groups ................................................................ 56
  3.1.4.5.7 Attribute Groups ................................................. 56
3.1.4.6 EnsureWorkbookModel ............................................. 56
  3.1.4.6.1 Messages ............................................................ 56
    3.1.4.6.1.1 EnsureWorkbookModelSoapIn ........................... 57
    3.1.4.6.1.2 EnsureWorkbookModelSoapOut ........................ 57
3.1.4.6.2 Elements ........................................................................................................ 57
3.1.4.6.2.1 EnsureWorkbookModel ........................................................................ 57
3.1.4.6.2.2 EnsureWorkbookModelResponse ....................................................... 58
3.1.4.6.3 Complex Types ............................................................................................ 58
3.1.4.6.3.1 WorkbookModelInfo ......................................................................... 58
3.1.4.6.4 Simple Types .............................................................................................. 59
3.1.4.6.5 Attributes .................................................................................................. 59
3.1.4.6.6 Groups ......................................................................................................... 59
3.1.4.6.7 Attribute Groups ........................................................................................ 59
3.1.4.7 GetApiVersion ................................................................................................. 59
3.1.4.7.1 Messages .................................................................................................... 59
3.1.4.7.1.1 GetApiVersionSoapIn ........................................................................ 59
3.1.4.7.1.2 GetApiVersionSoapOut ..................................................................... 60
3.1.4.7.2 Elements .................................................................................................... 60
3.1.4.7.2.1 GetApiVersion ...................................................................................... 60
3.1.4.7.2.2 GetApiVersionResponse .................................................................... 60
3.1.4.7.3 Complex Types .......................................................................................... 60
3.1.4.7.4 Simple Types .............................................................................................. 60
3.1.4.7.5 Attributes .................................................................................................. 60
3.1.4.7.6 Groups ......................................................................................................... 61
3.1.4.7.7 Attribute Groups ........................................................................................ 61
3.1.4.8 GetCell ............................................................................................................ 61
3.1.4.8.1 Messages .................................................................................................... 62
3.1.4.8.1.1 GetCellSoapIn .................................................................................... 62
3.1.4.8.1.2 GetCellSoapOut .................................................................................. 62
3.1.4.8.2 Elements .................................................................................................... 62
3.1.4.8.2.1 GetCell ................................................................................................ 62
3.1.4.8.2.2 GetCellResponse ................................................................................ 63
3.1.4.8.3 Complex Types .......................................................................................... 63
3.1.4.8.4 Simple Types .............................................................................................. 63
3.1.4.8.5 Attributes .................................................................................................. 63
3.1.4.8.6 Groups ......................................................................................................... 63
3.1.4.8.7 Attribute Groups ........................................................................................ 63
3.1.4.9 GetCellA1 ....................................................................................................... 63
3.1.4.9.1 Messages .................................................................................................... 64
3.1.4.9.1.1 GetCellA1SoapIn ................................................................................ 64
3.1.4.9.1.2 GetCellA1SoapOut ............................................................................. 65
3.1.4.9.2 Elements .................................................................................................... 65
3.1.4.9.2.1 GetCellA1 ............................................................................................ 65
3.1.4.9.2.2 GetCellA1Response ........................................................................... 65
3.1.4.9.3 Complex Types .......................................................................................... 66
3.1.4.9.4 Simple Types .............................................................................................. 66
3.1.4.9.5 Attributes .................................................................................................. 66
3.1.4.9.6 Groups ......................................................................................................... 66
3.1.4.9.7 Attribute Groups ........................................................................................ 66
3.1.4.10 GetChartImageUrl ....................................................................................... 66
3.1.4.10.1 Messages .................................................................................................... 67
3.1.4.10.1.1 GetChartImageUrlSoapIn ................................................................ 67
3.1.4.10.1.2 GetChartImageUrlSoapOut ............................................................... 67
3.1.4.10.2 Elements .................................................................................................... 67
3.1.4.10.2.1 GetChartImageUrl ............................................................................ 67
3.1.4.10.2.2 GetChartImageUrlResponse ............................................................. 68
3.1.4.10.3 Complex Types ........................................................................................ 68
3.1.4.10.3.1 Size ..................................................................................................... 68
3.1.4.10.4 Simple Types .............................................................................................. 69
3.1.4.10.5 Attributes .................................................................................................. 69
3.1.4.10.6 Groups ......................................................................................................... 69
3.1.4.10.7 Attribute Groups ........................................................................................ 69
3.1.4.11 GetPublishedItemNames ................................................................. 69
  3.1.4.11.1 Messages ................................................................................. 69
    3.1.4.11.1.1 GetPublishedItemNamesSoapIn ............................................ 70
    3.1.4.11.1.2 GetPublishedItemNamesSoapOut ......................................... 70
  3.1.4.11.2 Elements ................................................................................. 70
    3.1.4.11.2.1 GetPublishedItemNames ..................................................... 70
    3.1.4.11.2.2 GetPublishedItemNamesResponse ........................................ 70
  3.1.4.11.3 Complex Types ........................................................................ 71
    3.1.4.11.3.1 ArrayOfWorkbookItem ......................................................... 71
    3.1.4.11.3.2 WorkbookItem .................................................................. 71
  3.1.4.11.4 Simple Types ........................................................................... 71
    3.1.4.11.4.1 ItemType ........................................................................... 72
  3.1.4.11.5 Attributes ............................................................................... 72
  3.1.4.11.6 Groups .................................................................................... 72
  3.1.4.11.7 Attribute Groups ................................................................. 72
  3.1.4.12 GetRange ..................................................................................... 75
    3.1.4.12.1 Messages ................................................................................. 73
    3.1.4.12.1.1 GetRangeSoapIn ................................................................. 74
    3.1.4.12.1.2 GetRangeSoapOut ............................................................... 74
    3.1.4.12.2 Elements ............................................................................... 74
    3.1.4.12.2.1 GetRange ........................................................................... 74
    3.1.4.12.2.2 GetRangeResponse ............................................................. 75
  3.1.4.12.3 Complex Types ........................................................................ 75
  3.1.4.12.4 Simple Types ........................................................................... 75
  3.1.4.12.5 Attributes ............................................................................... 75
  3.1.4.12.6 Groups .................................................................................... 75
  3.1.4.12.7 Attribute Groups ................................................................. 75
  3.1.4.13 GetRangeA1 .................................................................................. 75
    3.1.4.13.1 Messages ................................................................................. 77
    3.1.4.13.1.1 GetRangeA1SoapIn ............................................................. 77
    3.1.4.13.1.2 GetRangeA1SoapOut ........................................................... 77
    3.1.4.13.2 Elements ............................................................................... 77
    3.1.4.13.2.1 GetRangeA1 ....................................................................... 77
    3.1.4.13.2.2 GetRangeA1Response ......................................................... 78
  3.1.4.13.3 Complex Types ........................................................................ 78
  3.1.4.13.4 Simple Types ........................................................................... 78
  3.1.4.13.5 Attributes ............................................................................... 78
  3.1.4.13.6 Groups .................................................................................... 78
  3.1.4.13.7 Attribute Groups ................................................................. 78
  3.1.4.14 GetSessionInformation ............................................................ 78
    3.1.4.14.1 Messages ................................................................................. 79
    3.1.4.14.1.1 GetSessionInformationSoapIn ............................................. 79
    3.1.4.14.1.2 GetSessionInformationSoapOut .......................................... 79
    3.1.4.14.2 Elements ............................................................................... 79
    3.1.4.14.2.1 GetSessionInformation ...................................................... 80
    3.1.4.14.2.2 GetSessionInformationResponse ........................................ 80
  3.1.4.14.3 Complex Types ........................................................................ 80
  3.1.4.14.4 Simple Types ........................................................................... 80
  3.1.4.14.5 Attributes ............................................................................... 80
  3.1.4.14.6 Groups .................................................................................... 81
  3.1.4.14.7 Attribute Groups ................................................................. 81
  3.1.4.15 GetSheetNames ................................................................. 81
    3.1.4.15.1 Messages ................................................................................. 81
    3.1.4.15.1.1 GetSheetNamesSoapIn ....................................................... 81
    3.1.4.15.1.2 GetSheetNamesSoapOut ................................................... 82
    3.1.4.15.2 Elements ............................................................................... 82
    3.1.4.15.2.1 GetSheetNames ................................................................. 82
    3.1.4.15.2.2 GetSheetNamesResponse ................................................... 82
<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.4.15.3</td>
<td>Complex Types</td>
</tr>
<tr>
<td>3.1.4.15.3.1</td>
<td>ArrayOfSheetInfo</td>
</tr>
<tr>
<td>3.1.4.15.3.2</td>
<td>SheetInfo</td>
</tr>
<tr>
<td>3.1.4.15.4</td>
<td>Simple Types</td>
</tr>
<tr>
<td>3.1.4.15.4.1</td>
<td>SheetType</td>
</tr>
<tr>
<td>3.1.4.15.4.2</td>
<td>SheetVisibility</td>
</tr>
<tr>
<td>3.1.4.15.5</td>
<td>Attributes</td>
</tr>
<tr>
<td>3.1.4.15.6</td>
<td>Groups</td>
</tr>
<tr>
<td>3.1.4.15.7</td>
<td>Attribute Groups</td>
</tr>
<tr>
<td>3.1.4.16</td>
<td>GetWorkbook</td>
</tr>
<tr>
<td>3.1.4.16.1</td>
<td>Messages</td>
</tr>
<tr>
<td>3.1.4.16.1.1</td>
<td>GetWorkbookSoapIn</td>
</tr>
<tr>
<td>3.1.4.16.1.2</td>
<td>GetWorkbookSoapOut</td>
</tr>
<tr>
<td>3.1.4.16.2</td>
<td>Elements</td>
</tr>
<tr>
<td>3.1.4.16.2.1</td>
<td>GetWorkbook</td>
</tr>
<tr>
<td>3.1.4.16.2.2</td>
<td>GetWorkbookResponse</td>
</tr>
<tr>
<td>3.1.4.16.3</td>
<td>Complex Types</td>
</tr>
<tr>
<td>3.1.4.16.4</td>
<td>Simple Types</td>
</tr>
<tr>
<td>3.1.4.16.5</td>
<td>Attributes</td>
</tr>
<tr>
<td>3.1.4.16.6</td>
<td>Groups</td>
</tr>
<tr>
<td>3.1.4.16.7</td>
<td>Attribute Groups</td>
</tr>
<tr>
<td>3.1.4.17</td>
<td>GetWorkbookConnections</td>
</tr>
<tr>
<td>3.1.4.17.1</td>
<td>Messages</td>
</tr>
<tr>
<td>3.1.4.17.1.1</td>
<td>GetWorkbookConnectionsSoapIn</td>
</tr>
<tr>
<td>3.1.4.17.1.2</td>
<td>GetWorkbookConnectionsSoapOut</td>
</tr>
<tr>
<td>3.1.4.17.2</td>
<td>Elements</td>
</tr>
<tr>
<td>3.1.4.17.2.1</td>
<td>GetWorkbookConnections</td>
</tr>
<tr>
<td>3.1.4.17.2.2</td>
<td>GetWorkbookConnectionsResponse</td>
</tr>
<tr>
<td>3.1.4.17.3</td>
<td>Complex Types</td>
</tr>
<tr>
<td>3.1.4.17.4</td>
<td>Simple Types</td>
</tr>
<tr>
<td>3.1.4.17.5</td>
<td>Attributes</td>
</tr>
<tr>
<td>3.1.4.17.6</td>
<td>Groups</td>
</tr>
<tr>
<td>3.1.4.17.7</td>
<td>Attribute Groups</td>
</tr>
<tr>
<td>3.1.4.18</td>
<td>NewWorkbook</td>
</tr>
<tr>
<td>3.1.4.18.1</td>
<td>Messages</td>
</tr>
<tr>
<td>3.1.4.18.1.1</td>
<td>NewWorkbookSoapIn</td>
</tr>
<tr>
<td>3.1.4.18.1.2</td>
<td>NewWorkbookSoapOut</td>
</tr>
<tr>
<td>3.1.4.18.2</td>
<td>Elements</td>
</tr>
<tr>
<td>3.1.4.18.2.1</td>
<td>NewWorkbook</td>
</tr>
<tr>
<td>3.1.4.18.2.2</td>
<td>NewWorkbookResponse</td>
</tr>
<tr>
<td>3.1.4.18.3</td>
<td>Complex Types</td>
</tr>
<tr>
<td>3.1.4.18.4</td>
<td>Simple Types</td>
</tr>
<tr>
<td>3.1.4.18.5</td>
<td>Attributes</td>
</tr>
<tr>
<td>3.1.4.18.6</td>
<td>Groups</td>
</tr>
<tr>
<td>3.1.4.18.7</td>
<td>Attribute Groups</td>
</tr>
<tr>
<td>3.1.4.19</td>
<td>OpenWorkbook</td>
</tr>
<tr>
<td>3.1.4.19.1</td>
<td>Messages</td>
</tr>
<tr>
<td>3.1.4.19.1.1</td>
<td>OpenWorkbookSoapIn</td>
</tr>
<tr>
<td>3.1.4.19.1.2</td>
<td>OpenWorkbookSoapOut</td>
</tr>
<tr>
<td>3.1.4.19.2</td>
<td>Elements</td>
</tr>
<tr>
<td>3.1.4.19.2.1</td>
<td>OpenWorkbook</td>
</tr>
<tr>
<td>3.1.4.19.2.2</td>
<td>OpenWorkbookResponse</td>
</tr>
<tr>
<td>3.1.4.19.3</td>
<td>Complex Types</td>
</tr>
<tr>
<td>3.1.4.19.4</td>
<td>Simple Types</td>
</tr>
<tr>
<td>3.1.4.19.5</td>
<td>Attributes</td>
</tr>
<tr>
<td>3.1.4.19.6</td>
<td>Groups</td>
</tr>
<tr>
<td>3.1.4.19.7</td>
<td>Attribute Groups</td>
</tr>
<tr>
<td>3.1.4.20</td>
<td>OpenWorkbookEx</td>
</tr>
</tbody>
</table>
3.1.4.20.1 Messages ......................................................... 95
3.1.4.20.1.1 OpenWorkbookExSoapIn........................................... 95
3.1.4.20.1.2 OpenWorkbookExSoapOut ........................................ 95
3.1.4.20.2 Elements ....................................................... 95
3.1.4.20.2.1 OpenWorkbookEx ............................................. 96
3.1.4.20.2.2 OpenWorkbookExResponse ..................................... 96
3.1.4.20.3 Complex Types .............................................. 96
3.1.4.20.4 Simple Types .............................................. 96
3.1.4.20.5 Attributes ................................................... 96
3.1.4.20.6 Groups ..................................................... 96
3.1.4.20.7 Attribute Groups ........................................... 97
3.1.4.21 OpenWorkbookForEditing ..................................... 97
3.1.4.21.1 Messages ..................................................... 98
3.1.4.21.1.1 OpenWorkbookForEditingSoapIn ................................ 98
3.1.4.21.1.2 OpenWorkbookForEditingSoapOut ......................... 98
3.1.4.21.2 Elements ..................................................... 98
3.1.4.21.2.1 OpenWorkbookForEditing ................................... 99
3.1.4.21.2.2 OpenWorkbookForEditingResponse ....................... 99
3.1.4.21.3 Complex Types ........................................... 99
3.1.4.21.4 Simple Types ........................................... 100
3.1.4.21.5 Attributes ................................................. 100
3.1.4.21.6 Groups .................................................. 100
3.1.4.21.7 Attribute Groups ....................................... 100
3.1.4.22 Refresh ..................................................... 100
3.1.4.22.1 Messages .................................................. 100
3.1.4.22.1.1 RefreshSoapIn ............................................ 100
3.1.4.22.1.2 RefreshSoapOut .......................................... 101
3.1.4.22.2 Elements .................................................. 101
3.1.4.22.2.1 Refresh .................................................. 101
3.1.4.22.2.2 RefreshResponse ......................................... 101
3.1.4.22.3 Complex Types ........................................... 101
3.1.4.22.4 Simple Types ........................................... 102
3.1.4.22.5 Attributes ............................................... 102
3.1.4.22.6 Groups .................................................. 102
3.1.4.22.7 Attribute Groups ....................................... 102
3.1.4.23 RefreshDetached ........................................... 102
3.1.4.23.1 Messages .................................................. 102
3.1.4.23.1.1 RefreshDetachedSoapIn .................................. 102
3.1.4.23.1.2 RefreshDetachedSoapOut ................................. 103
3.1.4.23.2 Elements .................................................. 103
3.1.4.23.2.1 RefreshDetached ........................................ 103
3.1.4.23.2.2 RefreshDetachedResponse ............................... 103
3.1.4.23.3 Complex Types ........................................... 104
3.1.4.23.3.1 ArrayOfConnectionDetails ................................. 104
3.1.4.23.3.2 ArrayOfDetailedRefreshResult .......................... 104
3.1.4.23.3.3 ConnectionDetails ...................................... 104
3.1.4.23.3.4 DetailedRefreshResult .................................. 105
3.1.4.23.4 Simple Types ........................................... 105
3.1.4.23.5 Attributes ............................................... 105
3.1.4.23.6 Groups .................................................. 105
3.1.4.23.7 Attribute Groups ....................................... 105
3.1.4.24 RefreshEx .................................................. 105
3.1.4.24.1 Messages .................................................. 105
3.1.4.24.1.1 RefreshExSoapIn ........................................ 105
3.1.4.24.1.2 RefreshExSoapOut ...................................... 106
3.1.4.24.2 Elements .................................................. 106
3.1.4.24.2.1 RefreshEx ............................................... 106
3.1.4.24.2.2 RefreshExResponse ..................................... 107
3.1.4.24.3 Complex Types .................................................107
3.1.4.24.4 Simple Types .................................................107
3.1.4.24.5 Attributes .......................................................107
3.1.4.24.6 Groups ............................................................107
3.1.4.24.7 Attribute Groups ..............................................107
3.1.4.25 SaveWorkbook ....................................................107
3.1.4.25.1 Messages .........................................................108
  3.1.4.25.1.1 SaveWorkbookSoapIn ....................................108
  3.1.4.25.1.2 SaveWorkbookSoapOut ...................................108
3.1.4.25.2 Elements .......................................................108
  3.1.4.25.2.1 SaveWorkbook .............................................109
  3.1.4.25.2.2 SaveWorkbookResponse ...................................109
3.1.4.25.3 Complex Types ..............................................109
3.1.4.25.4 Simple Types .................................................109
3.1.4.25.5 Attributes .......................................................109
3.1.4.25.6 Groups ............................................................109
3.1.4.25.7 Attribute Groups ..............................................109
3.1.4.26 SaveWorkbookCopy ............................................110
3.1.4.26.1 Messages .........................................................110
  3.1.4.26.1.1 SaveWorkbookCopySoapIn ................................111
  3.1.4.26.1.2 SaveWorkbookCopySoapOut ................................111
3.1.4.26.2 Elements .......................................................111
  3.1.4.26.2.1 SaveWorkbookCopy .......................................111
  3.1.4.26.2.2 SaveWorkbookCopyResponse ..............................112
3.1.4.26.3 Complex Types ..............................................112
3.1.4.26.4 Simple Types .................................................112
3.1.4.26.5 Attributes .......................................................112
3.1.4.26.6 Groups ............................................................112
3.1.4.26.7 Attribute Groups ..............................................112
3.1.4.27 SaveWorkbookCopyDetached ..................................112
3.1.4.27.1 Messages .........................................................113
  3.1.4.27.1.1 SaveWorkbookCopyDetachedSoapIn ........................113
  3.1.4.27.1.2 SaveWorkbookCopyDetachedSoapOut ........................113
3.1.4.27.2 Elements .......................................................113
  3.1.4.27.2.1 SaveWorkbookCopyDetached ................................113
  3.1.4.27.2.2 SaveWorkbookCopyDetachedResponse ........................114
3.1.4.27.3 Complex Types ..............................................114
3.1.4.27.4 Simple Types .................................................114
3.1.4.27.5 Attributes .......................................................114
3.1.4.27.6 Groups ............................................................114
3.1.4.27.7 Attribute Groups ..............................................114
3.1.4.28 SetCalculationOptions .......................................114
3.1.4.28.1 Messages .........................................................115
  3.1.4.28.1.1 SetCalculationOptionsSoapIn ............................115
  3.1.4.28.1.2 SetCalculationOptionsSoapOut ............................115
3.1.4.28.2 Elements .......................................................115
  3.1.4.28.2.1 SetCalculationOptions ....................................116
  3.1.4.28.2.2 SetCalculationOptionsResponse ..........................116
3.1.4.28.3 Complex Types ..............................................116
3.1.4.28.4 Simple Types .................................................116
  3.1.4.28.4.1 WorkbookCalculation ......................................116
3.1.4.28.5 Attributes .......................................................117
3.1.4.28.6 Groups ............................................................117
3.1.4.28.7 Attribute Groups ..............................................117
3.1.4.29 SetCell ...........................................................117
3.1.4.29.1 Messages .........................................................118
  3.1.4.29.1.1 SetCellSoapIn ..............................................118
  3.1.4.29.1.2 SetCellSoapOut ..............................................118
1 Introduction

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

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

1.1 Glossary

This document uses the following terms:

active sheet: The sheet that is currently selected.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

data provider: A known data source that is specific to a target type and that provides data to a collector type.
**data table**: A range of cells that is designated to perform what-if analysis for formulas, based on various input values.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**named range**: See defined name.
**new workbook session**: A session (2) that is not based on an existing workbook file.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**session identifier**: A key that enables an application to make reference to a session.
**sheet**: (1) A part of an Excel workbook. There are four types of sheets: **worksheet**, macro sheet, dialog sheet, and **chart sheet**. Multiple sheets are stored together within a workbook.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### 1.2 References

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

#### 1.2.1 Normative References

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

1.2.2 Informative References


1.3 Overview

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

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

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

1.3.1 Published Items and Results Returned by this Protocol

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

1.4 Relationship to Other Protocols

This protocol uses the SOAP message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2-1/2007] and [SOAP1.2-2/2007]. It transmits those messages by using HTTP, as described in [RFC2616], or Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS), as described in [RFC2818].

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

![Diagram showing the underlying messaging and transport stack](image)

Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

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

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

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

• Supported transports: This protocol can be implemented by using transports that support sending SOAP messages, as described in section 2.1.

• Protocol versions: This protocol is not versioned.

• Capability negotiation: This protocol does not support capability negotiation.
1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.
2 Messages

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

2.1 Transport

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

Protocol messages MUST be formatted as specified either in [SOAP1.1] section 4 or in [SOAP1.2-1/2007] 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 [SOAP1.2-1/2007] section 5.4. The version of the SOAP fault returned MUST correspond to the version of SOAP used for the request WSDL message.

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses XML schema, as specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2], and WSDL, as specified in [WSDL].

2.2.1 Namespaces

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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>http</td>
<td><a href="http://schemas.xmlsoap.org/wsdl/http/">http://schemas.xmlsoap.org/wsdl/http/</a></td>
<td></td>
</tr>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>soap12</td>
<td><a href="http://schemas.xmlsoap.org/wsdl/soap12/">http://schemas.xmlsoap.org/wsdl/soap12/</a></td>
<td>[SOAP1.2/1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[SOAP1.2/2]</td>
</tr>
<tr>
<td>tns</td>
<td><a href="http://schemas.microsoft.com/office/excel/server/webservices">http://schemas.microsoft.com/office/excel/server/webservices</a></td>
<td></td>
</tr>
<tr>
<td>wsd1</td>
<td><a href="http://schemas.microsoft.org/wsd1/">http://schemas.microsoft.org/wsd1/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA1/2]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[XMLSCHEMA2/2]</td>
</tr>
</tbody>
</table>

2.2.2 Messages

This specification does not define any common WSDL message definitions.
2.2.3 Elements
This specification does not define any common XML schema element definitions.

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfAnyType</td>
<td>An array of anyType elements.</td>
</tr>
<tr>
<td>ArrayOfStatus</td>
<td>An array of status information.</td>
</tr>
<tr>
<td>ArrayOfString</td>
<td>An array of string elements.</td>
</tr>
<tr>
<td>RangeCoordinates</td>
<td>Specifies a range on a workbook.</td>
</tr>
<tr>
<td>Status</td>
<td>The status information returned by the protocol server.</td>
</tr>
<tr>
<td>TypeExporter</td>
<td>MUST NOT be used.</td>
</tr>
</tbody>
</table>

2.2.4.1 ArrayOfAnyType

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

An array of anyType elements.

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

anyType: An anyType value.

2.2.4.2 ArrayOfStatus

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

An array of status information.

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

This complex type is defined as follows.

```xml
<xs:complexType name="ArrayOfStatus" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Status" nillable="true" type="tns:Status"/>
    </xs:sequence>
</xs:complexType>
```
Status: The status information returned by the protocol server.

2.2.4.3 ArrayOfString

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

An array of string elements.

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

string: A string value.

2.2.4.4 RangeCoordinates

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

Specifies a range on a workbook.

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

This complex type is defined as follows.

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

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

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

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

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

2.2.4.5 Status

Namespace: http://schemas.microsoft.com/office/excel/server/webservices
The status information returned by the protocol server.

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>The status is of severity error.</td>
</tr>
<tr>
<td>Warning</td>
<td>The status is of severity warning.</td>
</tr>
<tr>
<td>Information</td>
<td>The status is of severity information.</td>
</tr>
</tbody>
</table>

**Message:** The message description.

### 2.2.4.6 TypeExporter

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

MUST NOT be used.

It is defined as follows.

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

**CellError:** MUST NOT be used.

### 2.2.5 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CellError</td>
<td>The CellError type is an enumeration of a set of cell error values that a worksheet cell can contain.</td>
</tr>
<tr>
<td>RangeReference</td>
<td>A string that is used by the protocol server to identify a range on a sheet (2) in a workbook, in some of the operations specified by this protocol.</td>
</tr>
</tbody>
</table>
2.2.5.1 CellError

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

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

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

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

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

2.2.5.2 RangeReference

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

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

It has the following specification:
• MUST NOT be longer than 32768 characters.
• SHOULD use the A1 reference style as defined in [MSFT-ACRR], but MAY use any format supported by the server to identify the range.

This simple type is defined as follows.

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

### 2.2.5.3 SaveOptions

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

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>AllowOverwrite</td>
<td>If this flag is not set, then if the <code>workbookPath</code> specified in the <code>SaveWorkbookCopy</code> operation corresponds to an existing file, the protocol server MUST NOT overwrite the existing file.</td>
</tr>
<tr>
<td>LockAndCompareVersionsOnSave</td>
<td>MUST NOT be used.</td>
</tr>
</tbody>
</table>

### 2.2.5.4 SheetName

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

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

It has the following specification:

• MUST NOT be longer than 31 characters.
• If empty, the protocol server MUST use the first sheet (1) in the workbook unless specified otherwise in the operation.
If no sheet (1) in the workbook matches the requested **SheetName** simple type, the protocol server **MUST** return a SOAP fault response, which **SHOULD** contain the **InvalidSheetName** subcode but **MAY** contain any other subcode.

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

This simple type is defined as follows.

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

### 2.2.5.5 WorkbookType

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

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullWorkbook</td>
<td>The protocol server returns the workbook.</td>
</tr>
<tr>
<td>FullSnapshot</td>
<td>The protocol server returns a <strong>snapshot</strong> that contains the entire contents of the workbook.</td>
</tr>
<tr>
<td>PublishedItemsSnapshot</td>
<td>The protocol server returns a <strong>published items snapshot</strong>.</td>
</tr>
</tbody>
</table>

### 2.2.6 Attributes

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

### 2.2.7 Groups

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

### 2.2.8 Attribute Groups

This specification does not define any common **XML schema** attribute group definitions.
2.2.9 Common Data Structures

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

If [SOAP1.2-1/2007] or [SOAP1.2-2/2007] is used for transport, the protocol server MUST use the Value element inside the Subcode element to pass the error code, as specified in [SOAP1.2-1/2007], Section 5.4.1.3. Expected subcode values are specified in each operation. Common expected subcodes are specified in the Common List of Subcodes.

2.2.9.1 Faults

The following table describes the error codes.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>InternalError</td>
<td>An unexpected error occurred in the protocol server.</td>
</tr>
<tr>
<td>NotMemberOfRole</td>
<td>The user does not have permissions to complete the operation.</td>
</tr>
<tr>
<td>EncryptionTooLow</td>
<td>The protocol server is configured to accept only HTTPS, and the protocol client has tried to connect with a non HTTPS.</td>
</tr>
<tr>
<td>ConfirmDataRefresh</td>
<td>A <strong>connected data refresh confirmation</strong> is requested by the protocol client before connected data refresh on a workbook.</td>
</tr>
<tr>
<td>ExternalDataRefreshFailed</td>
<td>Connected data refresh failed.</td>
</tr>
<tr>
<td>ExternalDataRefreshFailedReal</td>
<td>Connected data refresh failed.&lt;1&gt;</td>
</tr>
<tr>
<td>ExternalDataRefreshFailedRealReadOrViewOnly</td>
<td>The query of one of the external sources failed (error message for users with read- or view-only permissions).</td>
</tr>
<tr>
<td>NoAccessToOdcReadOrViewOnly</td>
<td>The query of one of the external sources failed due to no access to the file in the format specified by [MS-ODCFF] (error message for users with read- or view-only permissions).</td>
</tr>
<tr>
<td>ErrorConnectingToDataSourceReadOrViewOnly</td>
<td>The query of one of the external sources failed due to failure to connect to the data source (error message for users with read- or view-only permissions).</td>
</tr>
<tr>
<td>InvalidUserNameOrPasswordReadOrViewOnly</td>
<td>The query of one of the external sources failed due to invalid credentials (error message for users with read- or view-only permissions).</td>
</tr>
<tr>
<td>InvalidUserNameOrPassword</td>
<td>The query of one of the external sources failed due to invalid credentials.</td>
</tr>
<tr>
<td>ExternalDataNotAllowedOnTL</td>
<td>The query of one of the external sources failed because external data connections are not allowed on the trusted location of the workbook session.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InvalidUriOrLocalConnectionPath</td>
<td>The query of one of the external sources failed due to a local path for the file in the format specified by [MS-ODCFF].</td>
</tr>
<tr>
<td>DclNotTrusted</td>
<td>The query of one of the external sources failed because of the security settings on the protocol server.</td>
</tr>
<tr>
<td>OdcFileDoesNotExist</td>
<td>The query of one of the external sources failed because the specified file in the format specified by [MS-ODCFF] does not exist.</td>
</tr>
<tr>
<td>ErrorAccessingSsoFromEcs</td>
<td>The query of one of the external sources failed because the protocol server was unable to resolve the credentials of the protocol client.</td>
</tr>
<tr>
<td>SsoError</td>
<td>The query of one of the external sources failed because the protocol server was unable to resolve the credentials of the protocol client.</td>
</tr>
<tr>
<td>ErrorConnectingToDataSource</td>
<td>The query of one of the external sources failed due to the inability to connect to the specified data source.</td>
</tr>
<tr>
<td>UnattendedAccountNotConfigured</td>
<td>The query of one of the external sources failed because the protocol server was unable to authenticate against the data source.</td>
</tr>
<tr>
<td>IncorrectAuthUsed</td>
<td>The query of one of the external sources failed due to a credentials delegation error.</td>
</tr>
<tr>
<td>ExternalDataRefreshFailed_Details</td>
<td>Detail information regarding the connection failure.</td>
</tr>
<tr>
<td>PivotNoOverlapAllowed</td>
<td>A PivotTable grew to the point where it would overlap another PivotTable.</td>
</tr>
<tr>
<td>PivotOverlapList</td>
<td>A PivotTable grew to the point where it would overlap a list.</td>
</tr>
<tr>
<td>PivotCubeNotUnique</td>
<td>An OLAP query found more than one cube of the same name.</td>
</tr>
<tr>
<td>NoDataItemsOlap</td>
<td>An OLAP query found nothing to display.</td>
</tr>
<tr>
<td>OlapSchemaFail</td>
<td>An OLAP query was unable to fetch the schema of the cube.</td>
</tr>
<tr>
<td>InvalidPivot</td>
<td>A PivotTable is invalid.</td>
</tr>
<tr>
<td>OlapNoMoreDetail</td>
<td>A PivotTable has reached the lowest level of detail.</td>
</tr>
<tr>
<td>DrillActionInvalid</td>
<td>A PivotTable member cannot be expanded any further.</td>
</tr>
<tr>
<td>DriverNotOlap</td>
<td>The database does not support OLAP.</td>
</tr>
<tr>
<td>CubeMissing</td>
<td>The OLAP query was unable to locate the cube in the database.</td>
</tr>
<tr>
<td>SkipLine</td>
<td>Not all data received from the OLAP query could be displayed.</td>
</tr>
<tr>
<td>CDimMismatch</td>
<td>The OLAP data provider returned an unexpected number of dimensions.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DataSetAxisTooBig</td>
<td>The <strong>row</strong> or <strong>column</strong> area of the PivotTable is too large.</td>
</tr>
<tr>
<td>CubeIntegrityError</td>
<td>There was a problem with the structure of the OLAP cube.</td>
</tr>
<tr>
<td>PivotNotExpandedtoProtectCellContents</td>
<td>The PivotTable would have filled <strong>worksheet cells</strong> that already contain values, so the operation was canceled.</td>
</tr>
<tr>
<td>PivotTruncatedSheetEdge</td>
<td>The PivotTable was truncated by a worksheet edge.</td>
</tr>
<tr>
<td>DdlGrpNotSup</td>
<td>The PivotTable operation is not supported by the OLAP data provider.</td>
</tr>
<tr>
<td>PivotFormulasCollided</td>
<td>Some <strong>calculated</strong> PivotTable members were removed from the PivotTable because new PivotTable members of the same name appeared during refresh.</td>
</tr>
<tr>
<td>NoRetItems</td>
<td>The query of the <strong>PivotTable page field</strong> did not return any items.</td>
</tr>
<tr>
<td>OlapConnectVisualModeProvider</td>
<td>The PivotTable operation is not supported by the OLAP data provider.</td>
</tr>
<tr>
<td>PivotNoSourceData</td>
<td>The PivotTable was saved without the underlying data.</td>
</tr>
<tr>
<td>PivotFilteringError</td>
<td>The PivotTable could not be filtered.</td>
</tr>
<tr>
<td>PivotTableIncorrectDateFormat</td>
<td>An invalid date was used to <strong>filter</strong> the PivotTable.</td>
</tr>
<tr>
<td>PivotTableEndingLargerThanStarting</td>
<td>An invalid pair of starting and ending values was used to filter the PivotTable.</td>
</tr>
<tr>
<td>PivotTableIncorrectNumberFormat</td>
<td>A non-numeric value was used to filter the PivotTable, when a numeric value was required.</td>
</tr>
<tr>
<td>PivotTableOutOfRange</td>
<td>An out of range value was used to filter the PivotTable.</td>
</tr>
<tr>
<td>MergedCellOverlap</td>
<td>The PivotTable could not <strong>grow</strong> over part of a <strong>merged cell</strong>.</td>
</tr>
<tr>
<td>PivotSourceNotAvailable</td>
<td>The PivotTable could not be refreshed because its source is missing.</td>
</tr>
<tr>
<td>UnexpectedPivotError</td>
<td>The PivotTable has encountered an unexpected error.</td>
</tr>
<tr>
<td>EditModeCannotAcquireLockUnknownReason</td>
<td>An attempt to acquire or refresh a lock on a file failed.</td>
</tr>
<tr>
<td>EditModeCannotAcquireLock</td>
<td>An attempt to acquire or refresh a lock on a file failed because the file is already <strong>locked</strong>.</td>
</tr>
<tr>
<td>EditModeCannotAcquireLockUnknownUser</td>
<td>An attempt to acquire or refresh a lock on a file failed because the file is locked by an unknown user.</td>
</tr>
<tr>
<td>EditModeWorkbookCheckedOutRequired</td>
<td>An attempt to acquire or refresh a lock on a file failed because the file is not <strong>checked out</strong>.</td>
</tr>
<tr>
<td>EditModeWorkbookCheckedOutToLocalDrafts</td>
<td>An attempt to acquire or refresh a lock on a file failed because the file is checked out to a local drafts folder.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SiteTooBusy</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SiteCollectionMemoryQuotaExceeded</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SessionMemoryQuotaExceeded</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SiteCollectionChartQuotaExceeded</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>RequestLimitsExceeded</td>
<td>Too many cells requested.</td>
</tr>
<tr>
<td>SavingUnsupportedInViewMode</td>
<td>An attempt was made to save a workbook from a <strong>view session</strong>.</td>
</tr>
<tr>
<td>SavingWorkbookNoWorkbookName</td>
<td>An attempt was made to save a workbook from a <strong>new workbook session</strong>.</td>
</tr>
<tr>
<td>FileSaveGeneric</td>
<td>The save operation failed.</td>
</tr>
<tr>
<td>SavingWorkbookInvalidPermissions</td>
<td>The user does not have permissions to save to the specified location.</td>
</tr>
<tr>
<td>SavingWorkbookMaximumSizeExceeded</td>
<td>An attempt to save a workbook failed because the workbook exceeds the maximum permitted size.</td>
</tr>
<tr>
<td>SavingWorkbookLockedOrCheckedOut</td>
<td>An attempt to save a workbook failed because the file is locked or checked out.</td>
</tr>
<tr>
<td>SavingWorkbookNotCheckedOut</td>
<td>An attempt to save a workbook failed because the file is not checked out.</td>
</tr>
<tr>
<td>SavingWorkbookAccessDenied</td>
<td>An attempt to save a workbook failed because the user does not have permission to overwrite an existing file.</td>
</tr>
<tr>
<td>SavingWorkbookInvalidFileType</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SavingWorkbookFileNameExists</td>
<td>An attempt to save a workbook failed because the file already exists and the option to overwrite existing files was not specified.</td>
</tr>
<tr>
<td>SavingWorkbookNotTrustedLocation</td>
<td>An attempt to save a workbook failed because of the security settings on the protocol server.</td>
</tr>
<tr>
<td>SavingWorkbookInvalidLocation</td>
<td>An attempt to save a workbook failed because the specified file location is invalid.</td>
</tr>
<tr>
<td>SavingWorkbookInvalidFileName</td>
<td>An attempt to save a workbook failed because the specified file name is invalid.</td>
</tr>
<tr>
<td>SavingWorkbookTrustedLocationSettingWarning</td>
<td>An attempt to save a workbook failed because of the security settings on the protocol server.</td>
</tr>
<tr>
<td>SavingWorkbookAlmostTooLarge</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SavingWorkbookTooLarge</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>NewWorkbookInvalidLocation</td>
<td>An attempt to create a new workbook failed because the specified location is invalid.</td>
</tr>
<tr>
<td>FileOpenSecuritySettings</td>
<td>Cannot open workbook because of security settings.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FileOpenNotFound</td>
<td>The specified workbook file does not exist.</td>
</tr>
<tr>
<td>FileOpenAccessDenied</td>
<td>The user does not have access to the workbook file.</td>
</tr>
<tr>
<td>FileCorrupt</td>
<td>The workbook file is in a format not recognized by the protocol server.</td>
</tr>
<tr>
<td>OpenWorkbookTooLarge</td>
<td>The protocol server failed to create a session for a workbook because it is too large.</td>
</tr>
<tr>
<td>OpenWorkbookLocationNotTrusted</td>
<td>The protocol server failed to create a session for a workbook because of the security settings on the protocol server.</td>
</tr>
<tr>
<td>OpenWorkbookAccessDenied</td>
<td>The protocol server failed to create a session for a workbook because the user does not have sufficient permissions.</td>
</tr>
<tr>
<td>WorkbookNotSupported</td>
<td>The protocol server failed to create a session for this workbook because it has unsupported content.</td>
</tr>
<tr>
<td>WorkbookNotSupportedViewRight</td>
<td>The protocol server failed to create a session for this workbook because it has unsupported content and the user does not have Open Item permission.</td>
</tr>
<tr>
<td>UnsupportedFeatureViewWarning</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>UnsupportedFeatureSaveCopyWarning</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ExternalWorkbookLinksDisabled</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>WorkbookLockedRead</td>
<td>An attempt to open a workbook failed because a file lock prevents it from being read.</td>
</tr>
<tr>
<td>EditModeCannotEditNonSharePointWorkbook</td>
<td>An attempt to open an edit session failed because the file is not in a document library.</td>
</tr>
<tr>
<td>EditModeAccessDenied</td>
<td>An attempt to open a workbook in an edit session failed because the user does not have sufficient permissions to edit the workbook file.</td>
</tr>
<tr>
<td>GenericFileOpenError</td>
<td>An unspecified error occurred when opening the workbook file.</td>
</tr>
<tr>
<td>IRMedWorkbook</td>
<td>An attempt to open a workbook failed because the workbook is in an Information Rights Management (IRM) document library.</td>
</tr>
<tr>
<td>ParameterInvalid</td>
<td>An attempt to open a workbook failed because the workbook contains one or more invalid workbook parameters.</td>
</tr>
<tr>
<td>ParameterLimitExceeded</td>
<td>An attempt to open a workbook failed because the workbook contains too many workbook parameters.</td>
</tr>
<tr>
<td>WorkbookInternalLimit</td>
<td>The workbook cannot be loaded by the protocol server because of a file-size limitation.</td>
</tr>
<tr>
<td>OldFormatConversionFailed</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EditModeWorkbookExternalData</td>
<td>An attempt to open a workbook as an edit session failed because it contains connected data or <em>user-defined functions (UDFs)</em>, and the workbook file has not been checked out.</td>
</tr>
<tr>
<td>FormulaEditingNotEnabled</td>
<td>The operation failed because it tried to set a <em>formula</em> into a workbook parameter.</td>
</tr>
<tr>
<td>DimensionAndArrayMismatch</td>
<td>An attempt to set a <em>range</em> failed because the dimensions of the specified array of cells do not match the specified range.</td>
</tr>
<tr>
<td>SetRangeFailure</td>
<td>The operation failed to set a value into a cell.</td>
</tr>
<tr>
<td>ObjectTypeNotSupported</td>
<td>The operation failed because it tried to update a cell with an object of an unsupported type.</td>
</tr>
<tr>
<td>ParameterDoesNotExist</td>
<td>The specified workbook parameter does not exist.</td>
</tr>
<tr>
<td>FormulaParseFailed</td>
<td>The specified formula could not be parsed.</td>
</tr>
<tr>
<td>SetParameterFailure</td>
<td>An unspecified error occurred when attempting to set a workbook parameter.</td>
</tr>
<tr>
<td>CannotEditPivotTable</td>
<td>The operation failed because it tried to set a cell in a PivotTable.</td>
</tr>
<tr>
<td>InsDelCellsNonblankCells</td>
<td>The operation failed because it would require shifting non-blank cells off the <em>sheet</em>.</td>
</tr>
<tr>
<td>InsDelCellsArrayFmla</td>
<td>The operation failed because it tried to change part of an <em>array formula</em>.</td>
</tr>
<tr>
<td>InsDelCellsPivotTable</td>
<td>The operation failed because it tried to change part of a PivotTable.</td>
</tr>
<tr>
<td>InsDelCellsTable</td>
<td>The operation failed because it tried to change part of a <em>table</em>.</td>
</tr>
<tr>
<td>InsDelCellsMergedCells</td>
<td>The operation failed because it tried to change part of a merged cell.</td>
</tr>
<tr>
<td>InsDelCellsDatatable</td>
<td>The operation failed because it tried to change part of a <em>data table</em>.</td>
</tr>
<tr>
<td>InsDelCellsRowsForWholeColumn</td>
<td>The operation failed because it tried to insert or delete a whole row, but the specified range is a whole column.</td>
</tr>
<tr>
<td>InsDelCellsColumnsForWholeRow</td>
<td>The operation failed because it tried to insert or delete a whole column, but the specified range is a whole row.</td>
</tr>
<tr>
<td>InvalidSheetName</td>
<td>The operation failed because the specified sheet (1) name does not exist.</td>
</tr>
<tr>
<td>RangeParseError</td>
<td>The operation failed because the specified range is invalid.</td>
</tr>
<tr>
<td>SpecifiedRangeNotFound</td>
<td>The operation failed because the specified range was not found.</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
<td>The operation failed because the specified range is not a valid sheet (2) location.</td>
</tr>
<tr>
<td>DiscontiguousRangeNotSupported</td>
<td>The operation failed because a <em>non-contiguous range</em> was specified.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
<td>The requested range was too large.</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
<td>The sheet (2) provided as the sheet (2) argument is not the same as the sheet (2) specified in the range argument.</td>
</tr>
<tr>
<td>ImageGenFailed</td>
<td>The protocol server failed to create an image.</td>
</tr>
<tr>
<td>ApiInvalidArgument</td>
<td>The operation failed because an argument that passed was invalid.</td>
</tr>
<tr>
<td>ApiInvalidCoordinate</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ConfirmationUnsupported</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ApiInvalidDimensionReturned</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ApiMissingSiteContext</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_IllegalPageSize</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_WorkbookNotSpecified</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_LoadProgress</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_FileCorrupt</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_WorkbookNotSupported</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_GenericFileOpenError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidParameterGeneric</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidParameterMandatoryMissing</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidParameterUnexpectedFormat</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidParameterOutOfRange</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidParameterOperationNotAllowed</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ParameterInvalid</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ParameterLimitExceeded</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ParameterDoesNotExist</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_SetParameterFailure</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Ewr_SetParameterFailureOnOpen</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ParameterReadFromDocPropsError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_FormulaEditingNotEnabled</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_InvalidVisibleItem</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_EditSaveAsOverCurrentWorkbook</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_RequestTimedOut</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ServerCommunicationError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Ewr_ServerAuthenticationError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Mewa_PageStateInitFailed</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Mewa_CriticalParameterMissing</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Mewa_GridViewRenderingFailed</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Mewa_FeatureNotActive</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>OperationNotSupportedInViewOnly</td>
<td>The operation failed because it is not supported in a view session.</td>
</tr>
<tr>
<td>OperationOnInvalidState</td>
<td>An attempt was made to open a session (2) that already has a workbook assigned.</td>
</tr>
<tr>
<td>InvalidOrTimedOutSession</td>
<td>The operation failed because the session (2) is invalid or timed out.</td>
</tr>
<tr>
<td>ScratchPadChangesLost</td>
<td>A view session that is not in view-only mode timed out while the user was attempting to convert it to an edit session.</td>
</tr>
<tr>
<td>MaxSessionsPerUserExceeded</td>
<td>The user has exceeded the maximum allowed session (2) count by the protocol server.</td>
</tr>
<tr>
<td>OperationCanceled</td>
<td>The user canceled a request that was being processed by a session (2).</td>
</tr>
<tr>
<td>MaxRequestDurationExceeded</td>
<td>The request failed because it took too long to process.</td>
</tr>
<tr>
<td>NoOccurrencesFound</td>
<td>The protocol server could not match the sought string to any worksheet cell relevant to the search.</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
<td>An invalid named object was requested.</td>
</tr>
<tr>
<td>NoPublishedItems</td>
<td>The workbook does not have any published items.</td>
</tr>
<tr>
<td>RetryError</td>
<td>The operation could not be finished at this time, but the request might succeed if it is sent again by the protocol client.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StopOpenOnRefreshFailure</td>
<td>An attempt to open a workbook failed because connected data refresh failed.</td>
</tr>
<tr>
<td>ProductExpired</td>
<td>The protocol server returns this error when the application <strong>license key</strong> is invalid.</td>
</tr>
<tr>
<td>SortActionInvalid</td>
<td>Sorting an <strong>AutoFilter</strong> failed because it contains merged cells that are not all the same size.</td>
</tr>
<tr>
<td>SortActionNotSupportedWithPivot</td>
<td>Sorting an AutoFilter failed because it contains a PivotTable.</td>
</tr>
<tr>
<td>CircularReferencesFound</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SessionNotFound</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>NewWorkbookSessionTimeout</td>
<td>A new workbook session has timed out.</td>
</tr>
<tr>
<td>EditXLSFile</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>UnsupportedFeatureEditWarning</td>
<td>Edit session is not supported for this workbook.</td>
</tr>
<tr>
<td>EditModeSessionExists</td>
<td>The view session could not be saved because the workbook is open in an edit session.</td>
</tr>
<tr>
<td>NewerWorkbookExists</td>
<td>The view session could not be saved because the workbook has been modified since the session (2) was opened.</td>
</tr>
<tr>
<td>ListEntryNotFound_1</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ListEntryNotFound_2</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ListEntryAlreadyExists_1</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ListEntryAlreadyExists_2</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>InvalidArgumentValue</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>MissingArgumentName</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>InvalidArgumentName</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ArgumentBadInRemove</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ManagementUIListEntryAlreadyExists</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ManagementUIListEntryNotFound</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ManagementUIInternalError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>UnsupportedUICulture</td>
<td>The <strong>UI culture</strong> is not supported by the protocol server.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UnsupportedDataCulture</td>
<td>The <strong>data culture</strong> is not supported by the protocol server.</td>
</tr>
<tr>
<td>SpecifiedItemNotFound</td>
<td>An attempt to set a range of cells in a workbook failed because the user does not have Open Item permission.</td>
</tr>
<tr>
<td>EcsSnapshotAccessDenied</td>
<td>The protocol server cannot return the workbook or a <strong>snapshot</strong>, because the user does not have Open Item permission.</td>
</tr>
<tr>
<td>SaveFailed</td>
<td>A save operation failed.</td>
</tr>
<tr>
<td>WrongEcs</td>
<td>An attempt was made to access an edit session that is owned by another instance of a protocol server in the <strong>farm</strong>.</td>
</tr>
<tr>
<td>PasteDataLimitHit</td>
<td>The string exceeds the maximum permitted length.</td>
</tr>
<tr>
<td>UndoRedoFail</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>CollabLocaleMismatch</td>
<td>The user joined an edit session whose data culture does not match the user’s data culture.</td>
</tr>
<tr>
<td>LocaleRestored</td>
<td>The data culture of an edit session has been restored to the user’s data culture.</td>
</tr>
<tr>
<td>InvalidEditSession</td>
<td>The edit session is invalid.</td>
</tr>
<tr>
<td>InvalidEditSessionId</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>UnableToLoadEditSession</td>
<td>Unable to load an edit session.</td>
</tr>
<tr>
<td>SavingWorkbookFailed</td>
<td>Failed to save workbook.</td>
</tr>
<tr>
<td>RefreshLockFailed</td>
<td>An attempt to refresh a lock on a workbook file failed.</td>
</tr>
<tr>
<td>InvalidChartName</td>
<td>An invalid chart name was specified.</td>
</tr>
<tr>
<td>RetryOutOfSync</td>
<td>The PivotTable operation failed.</td>
</tr>
<tr>
<td>OperationFailed</td>
<td>A workbook operation failed.</td>
</tr>
<tr>
<td>OpenInExcelActiveXError</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>Count</td>
<td>MUST NOT be used.</td>
</tr>
</tbody>
</table>

### 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** contain one of the following subcodes:
<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalError</td>
</tr>
<tr>
<td>OperationFailed</td>
</tr>
<tr>
<td>NotMemberOfRole</td>
</tr>
<tr>
<td>EncryptionTooLow</td>
</tr>
<tr>
<td>ConfirmDataRefresh</td>
</tr>
<tr>
<td>OperationOnInvalidState</td>
</tr>
<tr>
<td>InvalidOrTimedOutSession</td>
</tr>
<tr>
<td>MaxSessionsPerUserExceeded</td>
</tr>
<tr>
<td>OperationCanceled</td>
</tr>
<tr>
<td>MaxRequestDurationExceeded</td>
</tr>
<tr>
<td>RetryError</td>
</tr>
<tr>
<td>ProductExpired</td>
</tr>
<tr>
<td>NewWorkbookSessionTimeout</td>
</tr>
<tr>
<td>SpecifiedItemNotFound</td>
</tr>
<tr>
<td>CollabLocaleMismatch</td>
</tr>
<tr>
<td>InvalidEditSession</td>
</tr>
<tr>
<td>UnableToLoadEditSession</td>
</tr>
<tr>
<td>SpecifiedRangeNotFound</td>
</tr>
<tr>
<td>EditModeAccessDenied</td>
</tr>
<tr>
<td>EditModeCannotAcquireLockUnknownReason</td>
</tr>
<tr>
<td>RequestLimitsExceeded</td>
</tr>
<tr>
<td>SavingWorkbookFailed</td>
</tr>
<tr>
<td>RefreshLockFailed</td>
</tr>
<tr>
<td>ApiInvalidArgument</td>
</tr>
</tbody>
</table>
3 Protocol Details

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

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

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

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

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

![High-level sequence diagram](image)

**Figure 2: High-level sequence diagram**

**Session Behavior**

**View Sessions**

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

**New Workbook Sessions**

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

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

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

### Request Behavior

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

### 3.1 Server Details

#### 3.1.1 Abstract Data Model

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

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

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

#### 3.1.2 Timers

##### 3.1.2.1 Session Timeout

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

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing Rules

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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate</td>
<td>This operation is used to calculate a specific range in the workbook.</td>
</tr>
<tr>
<td>CalculateA1</td>
<td>This operation is used to calculate a specific range in the workbook.</td>
</tr>
<tr>
<td>CalculateWorkbook</td>
<td>This operation is used to calculate the workbook.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CancelRequest</td>
<td>This operation is used to cancel an operation that is currently running on the protocol server.</td>
</tr>
<tr>
<td>CloseWorkbook</td>
<td>This operation is used to close an existing session (2) or leave an edit session on the protocol server.</td>
</tr>
<tr>
<td>EnsureWorkbookModel</td>
<td>Loads workbook data model and returns information about the model.</td>
</tr>
<tr>
<td>GetApiVersion</td>
<td>This operation is used to get the version of the protocol server.</td>
</tr>
<tr>
<td>GetCell</td>
<td>This operation is used to retrieve the contents of a specific cell in the workbook.</td>
</tr>
<tr>
<td>GetCellA1</td>
<td>This operation is used to retrieve the contents of a specific cell in the workbook.</td>
</tr>
<tr>
<td>GetChartImageUrl</td>
<td>This operation returns the URL of a chart in a workbook.</td>
</tr>
<tr>
<td>GetPublishedItemNames</td>
<td>This operation returns a list of the published items in a workbook.</td>
</tr>
<tr>
<td>GetRange</td>
<td>This operation is used to retrieve the contents of a specific range in the workbook.</td>
</tr>
<tr>
<td>GetRangeA1</td>
<td>This operation is used to retrieve the contents of a named range in the workbook.</td>
</tr>
<tr>
<td>GetSessionInformation</td>
<td>This operation retrieves information about a session (2) on the protocol server.</td>
</tr>
<tr>
<td>GetSheetNames</td>
<td>The operation returns information about the sheets in a workbook.</td>
</tr>
<tr>
<td>GetWorkbook</td>
<td>This operation is used to retrieve a workbook file.</td>
</tr>
<tr>
<td>GetWorkbookConnections</td>
<td>Returns a list of data connection names in the workbook.</td>
</tr>
<tr>
<td>NewWorkbook</td>
<td>This operation is used to create a new workbook session.</td>
</tr>
<tr>
<td>OpenWorkbook</td>
<td>This operation is used to create a new view session, giving access to a workbook on the protocol server.</td>
</tr>
<tr>
<td>OpenWorkbookEx</td>
<td>This operation is used to create a new view session, giving access to a workbook on the protocol server.</td>
</tr>
<tr>
<td>OpenWorkbookForEditing</td>
<td>This operation is used to create a new edit session, giving access to a workbook on the protocol server.</td>
</tr>
<tr>
<td>Refresh</td>
<td>This operation is used to refresh connected data on a workbook.</td>
</tr>
<tr>
<td>RefreshDetached</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>RefreshEx</td>
<td>This operation is used to refresh connected data on a workbook.</td>
</tr>
<tr>
<td>SaveWorkbook</td>
<td>This operation is used to save a workbook associated with an open session (2) to the same location from which it was opened.</td>
</tr>
<tr>
<td>SaveWorkbookCopy</td>
<td>This operation is used to save a workbook associated with an open session (2) to a specified location, and in a specified format.</td>
</tr>
<tr>
<td>SaveWorkbookCopyDetached</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>SetCalculationOptions</td>
<td>This operation is used to set the workbook calculation mode.</td>
</tr>
<tr>
<td>SetCell</td>
<td>This operation is used to set a value to a specific worksheet cell in the workbook.</td>
</tr>
<tr>
<td>Operation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetCellA1</td>
<td>This operation is used to set a value to a specific worksheet cell in the workbook.</td>
</tr>
<tr>
<td>SetParameters</td>
<td>This operation sets values of <strong>workbook parameters</strong> in a workbook.</td>
</tr>
<tr>
<td>SetRange</td>
<td>This operation is used to set values to a range of worksheet cells on a workbook.</td>
</tr>
<tr>
<td>SetRangeA1</td>
<td>This operation is used to set values to a range of worksheet cells on a workbook.</td>
</tr>
</tbody>
</table>

### 3.1.4.1 Calculate

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

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

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

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

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
</tbody>
</table>
### 3.1.4.1.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateSoapIn</td>
<td>The request WSDL message for the <strong>Calculate</strong> WSDL operation.</td>
</tr>
<tr>
<td>CalculateSoapOut</td>
<td>The response WSDL message for the <strong>Calculate</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.1.1 CalculateSoapIn

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

The **SOAP action** value is:

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

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

#### 3.1.4.1.1.2 CalculateSoapOut

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

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

### 3.1.4.1.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate</td>
<td>The input data for the <strong>Calculate</strong> WSDL operation.</td>
</tr>
<tr>
<td>CalculateResponse</td>
<td>The result data for the <strong>Calculate</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.2.1 Calculate

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

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

---

**Subcode**

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>

---

[MS-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
<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>

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

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

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

### 3.1.4.1.2.2 CalculateResponse

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

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

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

### 3.1.4.1.3 Complex Types

None.

### 3.1.4.1.4 Simple Types

None.

### 3.1.4.1.5 Attributes

None.

### 3.1.4.1.6 Groups

None.

### 3.1.4.1.7 Attribute Groups

None.

### 3.1.4.2 CalculateA1

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

The following is the WSDL port type specification of the CalculateA1 WSDL operation.
The protocol client sends a **CalculateA1SoapIn** request message, and the protocol server responds with a **CalculateA1SoapOut** response message, as follows:

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>

### 3.1.4.2.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateA1SoapIn</td>
<td>The request WSDL message for the <strong>CalculateA1</strong> WSDL operation.</td>
</tr>
<tr>
<td>CalculateA1SoapOut</td>
<td>The response WSDL message for the <strong>CalculateA1</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.2.1.1 CalculateA1SoapIn
The request **WSDL message** for the **CalculateA1 WSDL operation**.

The **SOAP action** value is:

http://schemas.microsoft.com_office/excel/server/webservices/CalculateA1

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

### 3.1.4.2.1.2 CalculateA1SoapOut

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

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

### 3.1.4.2.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateA1</td>
<td>The input data for the <strong>CalculateA1 WSDL operation</strong>.</td>
</tr>
<tr>
<td>CalculateA1Response</td>
<td>The result data for the <strong>CalculateA1 WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.2.2.1 CalculateA1

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

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

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

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

- **rangeName**: The name of the range to calculate. MUST be present. MUST NOT be longer than 255 characters. See section 2.2.5.2.

#### 3.1.4.2.2.2 CalculateA1Response

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

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

[MS-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
<xs:complexType>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
</xs:complexType>

status: An array of status information as specified in section 2.2.4.2.

3.1.4.2.3 Complex Types
None.

3.1.4.2.4 Simple Types
None.

3.1.4.2.5 Attributes
None.

3.1.4.2.6 Groups
None.

3.1.4.2.7 Attribute Groups
None.

3.1.4.3 CalculateWorkbook
This operation is used to calculate the workbook.

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

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

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

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
</tbody>
</table>
### 3.1.4.3.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateWorkbookSoapIn</td>
<td>The request WSDL message for the <strong>CalculateWorkbook</strong> WSDL operation.</td>
</tr>
<tr>
<td>CalculateWorkbookSoapOut</td>
<td>The response WSDL message for the <strong>CalculateWorkbook</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.3.1.1 CalculateWorkbookSoapIn

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

The **SOAP action** value is:

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

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

#### 3.1.4.3.1.2 CalculateWorkbookSoapOut

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

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

### 3.1.4.3.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateWorkbook</td>
<td>The input data for the <strong>CalculateWorkbook</strong> WSDL operation.</td>
</tr>
<tr>
<td>CalculateWorkbookResponse</td>
<td>The result data for the <strong>CalculateWorkbook</strong> WSDL operation.</td>
</tr>
</tbody>
</table>
### 3.1.4.3.2.1 CalculateWorkbook

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

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

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

**calculateType:** The type of calculation to perform, as defined in section 3.1.4.3.4.1.

### 3.1.4.3.2.2 CalculateWorkbookResponse

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

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

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.3.3 Complex Types

None.

### 3.1.4.3.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalculateType</td>
<td>The type of calculation to be executed.</td>
</tr>
</tbody>
</table>

### 3.1.4.3.4.1 CalculateType

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

The type of calculation to be executed.

```xml
<xs:simpleType name="CalculateType" xmlns:xs="http://www.w3.org/2001/XMLSchema">
```
The following table specifies the allowable values for the **CalculateType** simple type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recalculate</td>
<td>The protocol server MUST <strong>calculate</strong> all formulas in the <strong>workbook</strong> that need to be calculated.</td>
</tr>
<tr>
<td>CalculateFull</td>
<td>The protocol server MUST calculate all formulas.</td>
</tr>
</tbody>
</table>

### 3.1.4.3.5 Attributes

None.

### 3.1.4.3.6 Groups

None.

### 3.1.4.3.7 Attribute Groups

None.

### 3.1.4.4 CancelRequest

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

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

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

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

- The protocol server SHOULD try to cancel all requests currently running on the protocol server against the **session (2)** specified in **sessionId**.

- The following MUST NOT cause an application error in the **CancelRequest** operation:
  - An operation is not currently running on the session (2).
  - A currently running operation on the session (2) cannot be cancelled.

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

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CancelRequestSoapIn</td>
<td>The request WSDL message for the <strong>CancelRequest</strong> WSDL operation.</td>
</tr>
<tr>
<td>CancelRequestSoapOut</td>
<td>The response WSDL message for the <strong>CancelRequest</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.4.1.1 CancelRequestSoapIn

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

The **SOAP action** value is:

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

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

### 3.1.4.4.1.2 CancelRequestSoapOut

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

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

### 3.1.4.4.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CancelRequest</td>
<td>The input data for the <strong>CancelRequest</strong> WSDL operation.</td>
</tr>
<tr>
<td>CancelRequestResponse</td>
<td>The result data for the <strong>CancelRequest</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.4.2.1 CancelRequest

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

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

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

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

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

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

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 CloseWorkbook

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseWorkbookSoapIn</td>
<td>The request WSDL message for the <strong>CloseWorkbook</strong> WSDL operation.</td>
</tr>
<tr>
<td>CloseWorkbookSoapOut</td>
<td>The response WSDL message for the <strong>CloseWorkbook</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.5.1.1 CloseWorkbookSoapIn

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

The **SOAP action** value is:

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

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

### 3.1.4.5.1.2 CloseWorkbookSoapOut

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

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

### 3.1.4.5.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseWorkbook</td>
<td>The input data for the <strong>CloseWorkbook</strong> WSDL operation.</td>
</tr>
<tr>
<td>CloseWorkbookResponse</td>
<td>The result data for the <strong>CloseWorkbook</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.5.2.1 CloseWorkbook

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

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

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

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

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

**status**: MUST NOT be present.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 EnsureWorkbookModel

Loads workbook data model and returns information about the model.

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

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

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

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

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureWorkbookModelSoapIn</td>
<td>The request WSDL message for the EnsureWorkbookModel WSDL operation.</td>
</tr>
<tr>
<td>EnsureWorkbookModelSoapOut</td>
<td>The response WSDL message for the EnsureWorkbookModel WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.6.1.1 EnsureWorkbookModelSoapIn

The request WSDL message for the EnsureWorkbookModel WSDL operation.

The SOAP action value is:

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

The SOAP body contains the EnsureWorkbookModel element.

### 3.1.4.6.1.2 EnsureWorkbookModelSoapOut

The response WSDL message for the EnsureWorkbookModel WSDL operation.

The SOAP body contains the EnsureWorkbookModelResponse element.

### 3.1.4.6.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnsureWorkbookModel</td>
<td>The input data for the EnsureWorkbookModel WSDL operation.</td>
</tr>
<tr>
<td>EnsureWorkbookModelResponse</td>
<td>The result data for the EnsureWorkbookModel WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.6.2.1 EnsureWorkbookModel

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

```
<xs:element name="EnsureWorkbookModel" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
sessionId: A session identifier. MUST be present. SHOULD have a value that has been obtained from a previous call to the OpenWorkbook, OpenWorkbookEx, OpenWorkbookForEditing, or NewWorkbook operations, but MAY instead be obtained in any other way.

3.1.4.6.2.2 EnsureWorkbookModelResponse

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

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

EnsureWorkbookModelResult: WorkbookModelInfo element as specified in section 3.1.4.6.3.1

status: An array of status information as specified in section 2.2.4.2.

3.1.4.6.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkbookModelInfo</td>
<td>Specifies server, database, version of the data model, and session timeout (section 3.1.2.1).</td>
</tr>
</tbody>
</table>

3.1.4.6.3.1 WorkbookModelInfo

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

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

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

Server: The server where the data model was loaded.

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

SessionInteractivityTimeout: Session timeout (section 3.1.2.1).

Version: Version of the data model.
3.1.4.6.4 Simple Types
None.

3.1.4.6.5 Attributes
None.

3.1.4.6.6 Groups
None.

3.1.4.6.7 Attribute Groups
None.

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetApiVersionSoapIn</td>
<td>The request WSDL message for the GetApiVersion WSDL operation.</td>
</tr>
<tr>
<td>GetApiVersionSoapOut</td>
<td>The response WSDL message for the GetApiVersion WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.7.1.1 GetApiVersionSoapIn
The request WSDL message for the GetApiVersion WSDL operation.

The SOAP action value is:

```
```

The SOAP body contains the GetApiVersion element.
3.1.4.7.1.2  GetApiVersionSoapOut

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

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

### 3.1.4.7.2  Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetApiVersion</td>
<td>The input data for the <strong>GetApiVersion WSDL operation</strong>.</td>
</tr>
<tr>
<td>GetApiVersionResponse</td>
<td>The result data for the <strong>GetApiVersion WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.7.2.1  GetApiVersion

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

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

#### 3.1.4.7.2.2  GetApiVersionResponse

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

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

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

**status**: MUST NOT be present.

#### 3.1.4.7.3  Complex Types

None.

#### 3.1.4.7.4  Simple Types

None.

#### 3.1.4.7.5  Attributes

None.

---

[MS-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
3.1.4.7.6 Groups
None.

3.1.4.7.7 Attribute Groups
None.

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

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

```xml
<wSDL:operation name="GetCell" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
</wSDL:operation>
```

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

1. The row, column on the worksheet specified by sheetName, specify the location of the sheet (2) cells to retrieve.

2. If formatted is false, the worksheet cell returned as the GetCellResult element MUST be either not present or one of the following types: boolean, double, string, or CellError (section 2.2.5.1).

3. If formatted is true, the worksheet cell returned as the GetCellResult element MUST be type string, and MUST be formatted using a number format according to how the worksheet cell is number formatted in the workbook.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>
3.1.4.8.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCellSoapIn</td>
<td>The request WSDL message for the GetCell WSDL operation.</td>
</tr>
<tr>
<td>GetCellSoapOut</td>
<td>The response WSDL message for the GetCell WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.8.1.1 GetCellSoapIn

The request WSDL message for the GetCell WSDL operation.

The SOAP action value is:

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

The SOAP body contains the GetCell element.

3.1.4.8.1.2 GetCellSoapOut

The response WSDL message for the GetCell WSDL operation.

The SOAP body contains the GetCellResponse element.

3.1.4.8.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCell</td>
<td>The input data for the GetCell WSDL operation.</td>
</tr>
<tr>
<td>GetCellResponse</td>
<td>The result data for the GetCell WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.8.2.1 GetCell

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

```xml
<xs:element name="GetCell" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
**sessionId**: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.

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

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

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

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

### 3.1.4.8.2.2 GetCellResponse

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

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

**GetCellResult**: The returned cell value.

**status**: An array of status information as specified in section **2.2.4.2**.

### 3.1.4.8.3 Complex Types

None.

### 3.1.4.8.4 Simple Types

None.

### 3.1.4.8.5 Attributes

None.

### 3.1.4.8.6 Groups

None.

### 3.1.4.8.7 Attribute Groups

None.

### 3.1.4.9 GetCellA1

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

The following is the **WSDL** port type specification of the **GetCellA1** **WSDL operation**.
The protocol client sends a `GetCellA1SoapIn` request message, and the protocol server responds with a `GetCellA1SoapOut` response message, as follows:

1. The `rangeName` specifies the location of the sheet (2) cell to retrieve. The sheet (2) that contains the sheet (2) cell to retrieve is determined in the following manner.
   - If `rangeName` specifies a sheet (2), that is the sheet (2) which MUST be used.
   - Otherwise, the sheet (2) specified in `sheetName` MUST be used, as specified in section 2.2.5.4.
   - If neither `sheetName` nor `rangeName` specify a sheet (2), the protocol server MAY use any sheet (1) in the workbook.

2. If `formatted` is false, the sheet (2) cell returned as the `GetCellA1Result` element MUST be either not present or one of the following types: `boolean`, `double`, `string`, or `CellError` (section 2.2.5.1).

3. If `formatted` is true, the sheet (2) cell returned as the `GetCellA1Result` element MUST be type `string`, and MUST be number formatted according to how the sheet (2) cell is number formatted in the workbook.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>

### 3.1.4.9.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.
<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCellA1SoapIn</td>
<td>The request WSDL message for the GetCellA1 WSDL operation.</td>
</tr>
<tr>
<td>GetCellA1SoapOut</td>
<td>The response WSDL message for the GetCellA1 WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.9.1.1 GetCellA1SoapIn

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

The **SOAP action** value is:

```
```

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

### 3.1.4.9.1.2 GetCellA1SoapOut

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

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

### 3.1.4.9.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCellA1</td>
<td>The input data for the <strong>GetCellA1</strong> WSDL operation.</td>
</tr>
<tr>
<td>GetCellA1Response</td>
<td>The result data for the <strong>GetCellA1</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.9.2.1 GetCellA1

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

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

**sessionId**: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.
**sheetName:** The name of the sheet (2) that contains the cell. See section 2.2.5.4 for specification. MUST be present.

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

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

### 3.1.4.9.2.2 GetCellA1Response

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

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

**GetCellA1Result:** The returned cell value.

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.9.3 Complex Types

None.

### 3.1.4.9.4 Simple Types

None.

### 3.1.4.9.5 Attributes

None.

### 3.1.4.9.6 Groups

None.

### 3.1.4.9.7 Attribute Groups

None.

### 3.1.4.10 GetChartImageUrl

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

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

```xml
<wsdl:operation name="GetChartImageUrl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="tns:GetChartImageUrlSoapIn"/>
    <wsdl:output message="tns:GetChartImageUrlSoapOut"/>
</wsdl:operation>
```
The protocol client sends a GetChartImageUrlSoapIn request message, and the protocol server responds with a GetChartImageUrlSoapOut response message as follows:

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

3.1.4.10.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetChartImageUrlSoapIn</td>
<td>The request WSDL message for the GetChartImageUrl WSDL operation.</td>
</tr>
<tr>
<td>GetChartImageUrlSoapOut</td>
<td>The response WSDL message for the GetChartImageUrl WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.10.1.1 GetChartImageUrlSoapIn

The request WSDL message for the GetChartImageUrl WSDL operation.

The SOAP action value is:

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

The SOAP body contains the GetChartImageUrl element.

3.1.4.10.1.2 GetChartImageUrlSoapOut

The response WSDL message for the GetChartImageUrl WSDL operation.

The SOAP body contains the GetChartImageUrlResponse element.

3.1.4.10.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetChartImageUrl</td>
<td>The input data for the GetChartImageUrl WSDL operation.</td>
</tr>
<tr>
<td>GetChartImageUrlResponse</td>
<td>The result data for the GetChartImageUrl WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.10.2.1 GetChartImageUrl

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

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

[MES-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
```
sessionId: A session identifier. MUST be present. SHOULD have a value that has been obtained from a previous call to the OpenWorkbook, OpenWorkbookEx, OpenWorkbookForEditing, or NewWorkbook operations, but MAY instead be obtained in any other way.

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

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

3.1.4.10.2.2 GetChartImageUrlResponse

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

GetChartImageUrlResult: The URL of the requested chart.

status: An array of Status elements as specified in section 2.2.4.2.

3.1.4.10.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Specifies the size of a chart image.</td>
</tr>
</tbody>
</table>

3.1.4.10.3.1 Size

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

Specifies the size of a chart image defined as follows.

```xml
<xs:complexType name="Size" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```
**Width:** Specifies the width of the image to be returned. MUST be an integer greater than or equal to 4 and less than or equal to 4096, or -1. If -1, the original width of the chart will be used. Must be -1 if **Height** is -1.

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

### 3.1.4.10.4 Simple Types
None.

### 3.1.4.10.5 Attributes
None.

### 3.1.4.10.6 Groups
None.

### 3.1.4.10.7 Attribute Groups
None.

### 3.1.4.11 GetPublishedItemNames

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

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

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

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

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

#### 3.1.4.11.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetPublishedItemNamesSoapIn</td>
<td>The request WSDL message for the GetPublishedItemNames WSDL operation.</td>
</tr>
<tr>
<td>GetPublishedItemNamesSoapOut</td>
<td>The response WSDL message for the GetPublishedItemNames WSDL operation.</td>
</tr>
</tbody>
</table>
3.1.4.11.1.1 GetPublishedItemNamesSoapIn
The request WSDL message for the GetPublishedItemNames WSDL operation.
The SOAP action value is:

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

The SOAP body contains the GetPublishedItemNames element.

3.1.4.11.1.2 GetPublishedItemNamesSoapOut
The response WSDL message for the GetPublishedItemNames WSDL operation.
The SOAP body contains the GetPublishedItemNamesResponse element.

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetPublishedItemNames</td>
<td>The input data for the GetPublishedItemNames WSDL operation.</td>
</tr>
<tr>
<td>GetPublishedItemNamesResponse</td>
<td>The result data for the GetPublishedItemNames WSDL operation.</td>
</tr>
</tbody>
</table>

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

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

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

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

```xml
<x:s:element name="GetPublishedItemNamesResponse" xmlns:x="http://www.w3.org/2001/XMLSchema">
  <x:s:complexType>
    <x:s:sequence>
      <x:s:element minOccurs="0" maxOccurs="1" name="GetPublishedItemNamesResult" type="tns:ArrayOfWorkbookItem"/>
      <x:s:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </x:s:sequence>
  </x:s:complexType>
</x:s:element>
```
GetPublishedItemNamesResult: An array of WorkbookItem elements as specified in section 3.1.4.11.3.1. MUST be present.

status: An array of Status elements as specified in section 2.2.4.2.

3.1.4.11.3  Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfWorkbookItem</td>
<td>An array of WorkbookItem elements.</td>
</tr>
<tr>
<td>WorkbookItem</td>
<td>Specifies the name and type of a published item.</td>
</tr>
</tbody>
</table>

3.1.4.11.3.1 ArrayOfWorkbookItem

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

An array of WorkbookItem elements defined as follows.

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

WorkbookItem: A WorkbookItem as specified in section 3.1.4.11.3.2.

3.1.4.11.3.2 WorkbookItem

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

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

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

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

Type: The type of the published item as specified in section 3.1.4.11.4.1.

3.1.4.11.4  Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ItemType</strong></td>
<td>Specifies the type of a published item.</td>
</tr>
</tbody>
</table>

#### 3.1.4.11.4.1 ItemType

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

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chart</td>
<td>Specifies that the published item is a chart.</td>
</tr>
<tr>
<td>PivotTable</td>
<td>Specifies that the published item is a PivotTable.</td>
</tr>
<tr>
<td>Table</td>
<td>Specifies that the published item is a list.</td>
</tr>
<tr>
<td>NamedRange</td>
<td>Specifies that the published item is a named range.</td>
</tr>
</tbody>
</table>

#### 3.1.4.11.5 Attributes

None.

#### 3.1.4.11.6 Groups

None.

#### 3.1.4.11.7 Attribute Groups

None.

#### 3.1.4.12 GetRange

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

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

```xml
<wsdl:operation name="GetRange" xmlns:wsdl="http://schemas.xmlsoap.org/wsd1/">
  <wsdl:input message="tns:GetRangeSoapIn"/>
  <wsdl:output message="tns:GetRangeSoapOut"/>
</wsdl:operation>
```
The protocol client sends a `GetRangeSoapIn` request message, and the protocol server responds with a `GetRangeSoapOut` response message, as follows:

1. The `rangeCoordinates` on the sheet (2) specified by `sheetName`, specify the location of the sheet (2) cells to retrieve.

2. Each element inside the `GetRangeResult` element MUST be of type `ArrayOfAnyType`. Each element inside `GetRangeResult` is a row of sheet (1) cells returned from the protocol server. The number of rows returned MUST be equal to the number of rows requested in the `rangeCoordinates` element.

3. Each element inside each of the `ArrayOfAnyType` type nested inside `GetRangeResult` MUST be a single workbook sheet (2) cell. The number of elements inside each such nested `ArrayOfAnyType` MUST be equal to the number of columns requested in the `rangeCoordinates` element.

4. If `formatted` is false, all sheet (2) cells returned as part of the `GetRangeResult` element MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: `Boolean`, `double`, `string`, or `CellError` (section 2.2.5.1).

5. If `formatted` is true, all sheet cells returned as part of the `GetRangeResult` element MUST be type `string`, and MUST be number formatted according to how the sheet (1) cell is number formatted in the workbook.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>

### 3.1.4.12.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRangeSoapIn</td>
<td>The request WSDL message for the GetRange WSDL operation.</td>
</tr>
<tr>
<td>Message</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetRangeSoapOut</td>
<td>The response WSDL message for the GetRange WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.12.1.1 GetRangeSoapIn

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

The **SOAP action** value is:

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

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

### 3.1.4.12.1.2 GetRangeSoapOut

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

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

### 3.1.4.12.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRange</td>
<td>The input data for the GetRange WSDL operation.</td>
</tr>
<tr>
<td>GetRangeResponse</td>
<td>The result data for the GetRange WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.12.2.1 GetRange

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

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

**sessionId**: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.
**sheetName**: The name of the sheet (2) that contains the range. See section 2.2.5.4 for specification. MUST be present. MUST NOT be empty.

**rangeCoordinates**: The range to retrieve.
- MUST be present.
- MUST NOT be empty, as specified in section 2.2.4.4.
- The product of **Height** and **Width** MUST NOT be greater than 1,000,000.

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

### 3.1.4.12.2.2 GetRangeResponse

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

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

**GetRangeResult**: The returned range value.

**status**: An array of status information as specified in section 2.2.4.2.

### 3.1.4.12.3 Complex Types

None.

### 3.1.4.12.4 Simple Types

None.

### 3.1.4.12.5 Attributes

None.

### 3.1.4.12.6 Groups

None.

### 3.1.4.12.7 Attribute Groups

None.

### 3.1.4.13 GetRangeA1

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

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

```
<wsdl:operation name="GetRangeA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetRangeA1SoapIn"/>
</wsdl:operation>
```
The protocol client sends a `GetRangeA1SoapIn` request message, and the protocol server responds with a `GetRangeA1SoapOut` response message, as follows:

- The `rangeName` specifies the named range of the cells on the **sheet (2)** to retrieve. The sheet (2) that contains the sheet (2) cells to retrieve is determined in the following manner:
  - If `rangeName` specifies a sheet (2), that is the sheet (2) that MUST be used.
  - Otherwise, the sheet (2) specified in `sheetName` MUST be used, as specified in section 2.2.5.4. If neither `sheetName` nor `rangeName` specify a sheet (2), the protocol server MAY use any sheet (2) in the workbook.

- Each element inside the `GetRangeA1Result` element MUST be of type `ArrayOfAnyType`. Each element inside the `GetRangeA1Result` is a row of sheet (1) **cells** returned from the protocol server. The number of rows MUST be equal to the rows in the named range specified by `rangeName`.

- Each element inside each of the `ArrayOfAnyType` type nested inside `GetRangeA1Result` MUST be a single workbook sheet (2) cell. The number of elements inside each such nested `ArrayOfAnyType` MUST be equal to the number of **columns** in the named range specified by `rangeName`.

- If `formatted` is false, all sheet (2) cells returned as part of the `GetRangeA1Result` element MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: `Boolean`, `double`, `string`, or `CellError` (section 2.2.5.1).

- If `formatted` is true, all sheet (2) cells returned as part of the `GetRangeA1Result` element MUST be type `string`, and MUST be number formatted according to how the sheet (2) cell is number formatted in the workbook.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscontiguousRangeNotSupported</td>
</tr>
<tr>
<td>EcsInvalidNamedObject</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>RangeDefinitionError</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
<tr>
<td>RangeRequestAreaExceeded</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
</tbody>
</table>
3.1.4.13.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRangeA1SoapIn</td>
<td>The request WSDL message for the GetRangeA1 WSDL operation.</td>
</tr>
<tr>
<td>GetRangeA1SoapOut</td>
<td>The response WSDL message for the GetRangeA1 WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.13.1.1 GetRangeA1SoapIn

The request WSDL message for the GetRangeA1 WSDL operation.

The SOAP action value is:


The SOAP body contains the GetRangeA1 element.

3.1.4.13.1.2 GetRangeA1SoapOut

The response WSDL message for the GetRangeA1 WSDL operation.

The SOAP body contains the GetRangeA1Response element.

3.1.4.13.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRangeA1</td>
<td>The input data for the GetRangeA1 WSDL operation.</td>
</tr>
<tr>
<td>GetRangeA1Response</td>
<td>The result data for the GetRangeA1 WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.13.2.1 GetRangeA1

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

```xml
<xs:element name="GetRangeA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
**sessionId:** A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the `OpenWorkbook`, `OpenWorkbookEx`, `OpenWorkbookForEditing`, or `NewWorkbook` operation, but MAY instead be obtained in any other way.

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

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

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

### 3.1.4.13.2.2 GetRangeA1Response

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

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

**GetRangeA1Result:** The returned range value.

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.13.3 Complex Types

None.

### 3.1.4.13.4 Simple Types

None.

### 3.1.4.13.5 Attributes

None.

### 3.1.4.13.6 Groups

None.

### 3.1.4.13.7 Attribute Groups

None.

### 3.1.4.14 GetSessionInformation

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

The following is the **WSDL** port type specification of the **GetSessionInformation WSDL operation**.
The protocol client sends a GetSessionInformationSoapIn request message, and the protocol server responds with a GetSessionInformationSoapOut response message, as follows:

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

### 3.1.4.14.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSessionInformationSoapIn</td>
<td>The request WSDL message for the GetSessionInformation WSDL operation.</td>
</tr>
<tr>
<td>GetSessionInformationSoapOut</td>
<td>The response WSDL message for the GetSessionInformation WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.14.1.1 GetSessionInformationSoapIn

The request WSDL message for the GetSessionInformation WSDL operation.

The SOAP action value is:

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

The SOAP body contains the GetSessionInformation element.

### 3.1.4.14.1.2 GetSessionInformationSoapOut

The response WSDL message for the GetSessionInformation WSDL operation.

The SOAP body contains the GetSessionInformationResponse element.

### 3.1.4.14.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSessionInformation</td>
<td>The input data for the GetSessionInformation WSDL operation.</td>
</tr>
</tbody>
</table>
### 3.1.4.14.2.1 GetSessionInformation

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

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

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

### 3.1.4.14.2.2 GetSessionInformationResponse

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

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

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

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

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

**status:** An array of `Status` elements as specified in section 2.2.4.2.

### 3.1.4.14.3 Complex Types

None.

### 3.1.4.14.4 Simple Types

None.

### 3.1.4.14.5 Attributes

None.
None.

3.1.4.14.6 Groups
None.

3.1.4.14.7 Attribute Groups
None.

3.1.4.15 GetSheetNames
The operation returns information about the sheets in a workbook.

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

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

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

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

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSheetNamesSoapIn</td>
<td>The request WSDL message for the GetSheetNames WSDL operation.</td>
</tr>
<tr>
<td>GetSheetNamesSoapOut</td>
<td>The response WSDL message for the GetSheetNames WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.15.1.1 GetSheetNamesSoapIn
The request WSDL message for the GetSheetNames WSDL operation.

The SOAP action value is:

The SOAP body contains the GetSheetNames element.

3.1.4.15.1.2 GetSheetNamesSoapOut

The response WSDL message for the GetSheetNames WSDL operation.

The SOAP body contains the GetSheetNamesResponse element.

3.1.4.15.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSheetNames</td>
<td>The input data for the GetSheetNames WSDL operation.</td>
</tr>
<tr>
<td>GetSheetNamesResponse</td>
<td>The result data for the GetSheetNames WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.15.2.1 GetSheetNames

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

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

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

3.1.4.15.2.2 GetSheetNamesResponse

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

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

GetSheetNamesResult: An array of SheetInfo elements as specified in section 3.1.4.15.3.1. MUST be present.

status: An array of status information as specified in section 2.2.4.2.

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfSheetInfo</td>
<td>An array of SheetInfo elements.</td>
</tr>
<tr>
<td>SheetInfo</td>
<td>Specifies the name, type, and visible condition of a sheet (1).</td>
</tr>
</tbody>
</table>

### 3.1.4.15.3.1 ArrayOfSheetInfo

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

An array of SheetInfo elements defined as follows.

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

**SheetInfo:** A SheetInfo as specified in section 3.1.4.15.3.2.

### 3.1.4.15.3.2 SheetInfo

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

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

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

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

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

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

### 3.1.4.15.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SheetType</td>
<td>Specifies the type of a sheet (1).</td>
</tr>
<tr>
<td>SheetVisibility</td>
<td>Specifies the visible condition of a sheet (1).</td>
</tr>
</tbody>
</table>
### 3.1.4.15.4.1 SheetType

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

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worksheet</td>
<td>A worksheet.</td>
</tr>
<tr>
<td>Chartsheet</td>
<td>A chart sheet.</td>
</tr>
</tbody>
</table>

### 3.1.4.15.4.2 SheetVisibility

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

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

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

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible</td>
<td>Specifies that the sheet (1) is visible.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Specifies that the sheet (1) is not visible, but can be unhidden via the client UI.</td>
</tr>
<tr>
<td>VeryHidden</td>
<td>Specifies that the sheet (1) is not visible and cannot be unhidden via the client UI.</td>
</tr>
</tbody>
</table>

### 3.1.4.15.5 Attributes

None.

### 3.1.4.15.6 Groups

None.

### 3.1.4.15.7 Attribute Groups
3.1.4.16  GetWorkbook

This operation is used to retrieve a workbook file.

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

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

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

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>EcsSnapshotAccessDenied</td>
</tr>
<tr>
<td>SaveFailed</td>
</tr>
</tbody>
</table>

3.1.4.16.1  Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWorkbookSoapIn</td>
<td>The request WSDL message for the GetWorkbook WSDL operation.</td>
</tr>
<tr>
<td>GetWorkbookSoapOut</td>
<td>The response WSDL message for the GetWorkbook WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.16.1.1  GetWorkbookSoapIn

The request WSDL message for the GetWorkbook WSDL operation.

The SOAP action value is:

http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook
The SOAP body contains the `<GetWorkbook>` element.

### 3.1.4.16.1.2 GetWorkbookSoapOut

The response WSDL message for the GetWorkbook WSDL operation.

The SOAP body contains the `<GetWorkbookResponse>` element.

### 3.1.4.16.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWorkbook</td>
<td>The input data for the GetWorkbook WSDL operation.</td>
</tr>
<tr>
<td>GetWorkbookResponse</td>
<td>The result data for the GetWorkbook WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.16.2.1 GetWorkbook

The `<GetWorkbook>` element specifies the input data for the GetWorkbook WSDL operation.

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

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

**workbookType:** The form of the requested workbook as specified in section 2.2.5.5.

#### 3.1.4.16.2.2 GetWorkbookResponse

The `<GetWorkbookResponse>` element specifies the result data for the GetWorkbook WSDL operation.

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

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

**status:** An array of status information as specified in section 2.2.4.2.
3.1.4.16.3  Complex Types
None.

3.1.4.16.4  Simple Types
None.

3.1.4.16.5  Attributes
None.

3.1.4.16.6  Groups
None.

3.1.4.16.7  Attribute Groups
None.

3.1.4.17  GetWorkbookConnections

Returns a list of data connection names in the workbook.

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

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

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

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

3.1.4.17.1  Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWorkbookConnectionsSoapIn</td>
<td>The request WSDL message for the GetWorkbookConnections WSDL operation.</td>
</tr>
<tr>
<td>GetWorkbookConnectionsSoapOut</td>
<td>The response WSDL message for the GetWorkbookConnections WSDL operation.</td>
</tr>
</tbody>
</table>
3.1.4.17.1.1 GetWorkbookConnectionsSoapIn

The request WSDL message for the GetWorkbookConnections WSDL operation.

The SOAP action value is:

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

The SOAP body contains the GetWorkbookConnections element.

3.1.4.17.1.2 GetWorkbookConnectionsSoapOut

The response WSDL message for the GetWorkbookConnections WSDL operation.

The SOAP body contains the GetWorkbookConnectionsResponse element.

3.1.4.17.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetWorkbookConnections</td>
<td>The input data for the GetWorkbookConnections WSDL operation.</td>
</tr>
<tr>
<td>GetWorkbookConnectionsResponse</td>
<td>The result data for the GetWorkbookConnections WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.17.2.1 GetWorkbookConnections

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

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

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

3.1.4.17.2.2 GetWorkbookConnectionsResponse

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

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

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

**status**: An array of status information as specified in section 2.2.4.2.

### 3.1.4.17.3 Complex Types

None.

### 3.1.4.17.4 Simple Types

None.

### 3.1.4.17.5 Attributes

None.

### 3.1.4.17.6 Groups

None.

### 3.1.4.17.7 Attribute Groups

None.

### 3.1.4.18 NewWorkbook

This operation is used to create a new workbook session.

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

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

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

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

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

3.1.4.18.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewWorkbookSoapIn</td>
<td>The request WSDL message for the NewWorkbook WSDL operation.</td>
</tr>
<tr>
<td>NewWorkbookSoapOut</td>
<td>The response WSDL message for the NewWorkbook WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.18.1.1 NewWorkbookSoapIn

The request WSDL message for the NewWorkbook WSDL operation.

The SOAP action value is:

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

The SOAP body contains the NewWorkbook element.

3.1.4.18.1.2 NewWorkbookSoapOut

The response WSDL message for the NewWorkbook WSDL operation.

The SOAP body contains the NewWorkbookResponse element.

3.1.4.18.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewWorkbook</td>
<td>The input data for the NewWorkbook WSDL operation.</td>
</tr>
<tr>
<td>NewWorkbookResponse</td>
<td>The result data for the NewWorkbook WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.18.2.1 NewWorkbook

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

```xml
<x:s:element name="NewWorkbook" xmlns:x="http://www.w3.org/2001/XMLSchema">
  <x:s:complexType>
    <x:s:sequence>
      <x:s:element minOccurs="0" maxOccurs="1" name="path" type="x:s:string"/>
      <x:s:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="x:s:string"/>
      <x:s:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="x:s:string"/>
    </x:s:sequence>
  </x:s:complexType>
</x:s:element>
```
path: The trusted location of the new workbook session. MUST be present. MUST NOT be empty. MUST NOT be longer than 4095 characters. MUST be a UNC, or a URI as specified in [RFC2396].

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

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

3.1.4.18.2.2 NewWorkbookResponse

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

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

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

status: The status information returned by the protocol server.

3.1.4.18.3 Complex Types

None.

3.1.4.18.4 Simple Types

None.

3.1.4.18.5 Attributes

None.

3.1.4.18.6 Groups

None.

3.1.4.18.7 Attribute Groups

None.

3.1.4.19 OpenWorkbook

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

The following is the WSDL port type specification of the OpenWorkbook WSDL operation.
The protocol client sends an **OpenWorkbookSoapIn** request message, and the protocol server responds with an **OpenWorkbookSoapOut** response message, as follows:

1. The workbook file, specified by **workbookPath**, is specified as follows:
   1. It **SHOULD** be in a format specified by [MS-XLSB], [ISO/IEC29500-1:2011], [MS-OFFMACRO2], or [ODF], but **MAY** be in any other format.
   2. If it is in the [ISO/IEC29500-1:2011] format:
      1. It **MAY** contain any valid SpreadsheetML records in that format.
      2. If a **odcFile** attribute of a **connection** element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.1) is present, it **SHOULD** be the path of a file in the format specified by [MS-ODCFF].
      3. If a **connection** attribute of **dbPr** element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.3) is present, it **SHOULD** be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].
   3. If it is in the [MS-XLSB] format:
      1. It **MAY** 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** fall back to some other supported language. **uiCultureName** **MAY** 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** fall back to some other supported language. **dataCultureName** **MAY** be empty, in which case the determination of which language to default to is protocol-server-implementation specific.
4. If the operation succeeds, the protocol server **MUST** return the **session identifier** of the newly created **session (2)** in the response message, to be used in any future operations in the newly created session (2).
5. If there was an application error on the protocol server during the operation, the protocol server **MUST** return a SOAP fault response, which **SHOULD** contain one of the following subcodes or any other subcodes specified in section 2.2.9.2, but **MAY** instead contain any subcode specified in section 2.2.9.1 or any other subcode.

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileOpenNotFound</td>
</tr>
</tbody>
</table>
### Subcode

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileOpenAccessDenied</td>
</tr>
<tr>
<td>FileCorrupt</td>
</tr>
<tr>
<td>GenericFileOpenError</td>
</tr>
<tr>
<td>WorkbookNotSupported</td>
</tr>
</tbody>
</table>

### 3.1.4.19.1  Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbookSoapIn</td>
<td>The request WSDL message for the OpenWorkbook <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>OpenWorkbookSoapOut</td>
<td>The response WSDL message for the OpenWorkbook <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

### 3.1.4.19.1.1  OpenWorkbookSoapIn

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

The **SOAP action** value is:

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

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

### 3.1.4.19.1.2  OpenWorkbookSoapOut

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

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

### 3.1.4.19.2  Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbook</td>
<td>The input data for the OpenWorkbook <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>OpenWorkbookResponse</td>
<td>The result data for the OpenWorkbook <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

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

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

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

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

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

### 3.1.4.19.2.2 OpenWorkbookResponse

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

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

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

**status**: The status information returned by the protocol server.

### 3.1.4.19.3 Complex Types

None.

### 3.1.4.19.4 Simple Types

None.

### 3.1.4.19.5 Attributes

None.

### 3.1.4.19.6 Groups

None.

### 3.1.4.19.7 Attribute Groups

None.
3.1.4.20 **OpenWorkbookEx**

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

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

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

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

3.1.4.20.1 **Messages**

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbookExSoapIn</td>
<td>The request WSDL message for the <strong>OpenWorkbookEx WSDL operation</strong>.</td>
</tr>
<tr>
<td>OpenWorkbookExSoapOut</td>
<td>The response WSDL message for the <strong>OpenWorkbookEx WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

3.1.4.20.1.1 **OpenWorkbookExSoapIn**

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

The **SOAP action** value is:

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

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

3.1.4.20.1.2 **OpenWorkbookExSoapOut**

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

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

3.1.4.20.2 **Elements**

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbookEx</td>
<td>The input data for the <strong>OpenWorkbookEx WSDL operation</strong>.</td>
</tr>
<tr>
<td>OpenWorkbookExResponse</td>
<td>The result data for the <strong>OpenWorkbookEx WSDL operation</strong>.</td>
</tr>
</tbody>
</table>
3.1.4.20.2.1 OpenWorkbookEx

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

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

workbookPath: See section 3.1.4.19.2.1.

uiCultureName: See section 3.1.4.19.2.1.

dataCultureName: See section 3.1.4.19.2.1.

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

3.1.4.20.2.2 OpenWorkbookExResponse

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

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

OpenWorkbookExResult: See section 3.1.4.19.2.2.

status: See section 3.1.4.19.2.2.

3.1.4.20.3 Complex Types

None.

3.1.4.20.4 Simple Types

None.

3.1.4.20.5 Attributes

None.

3.1.4.20.6 Groups

None.
3.1.4.20.7  Attribute Groups
None.

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

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

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

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

1. The workbook file, specified by workbookPath, is specified as follows:
   1. It SHOULD be in a format specified by [MS-XLSB], [ISO/IEC29500-1:2011], [MS-OFFMACRO2], or [ODF], but MAY be in any other format.
   2. If it is in the [ISO/IEC29500-1:2011] format:
      1. It MAY<18> contain any valid SpreadsheetML records in that format.
      2. If an odcFile attribute of a connection element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.1) is present, it SHOULD be the path of a file in the format specified by [MS-ODCFF].
      3. If a connection attribute of dbPr element ([ISO/IEC29500-1:2011], Part 1, Section 18.13.3) is present, it SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].
   3. If it is in the [MS-XLSB] format:
      1. It MAY<19> contain any valid records in that format.
      2. If a brtBeginExternalConnection record is present and it has the stConnectionFile field present, the stConnectionFile SHOULD be the path of a file in the format specified by [MS-ODCFF].
      3. If a BrtBeginECDbProps record is present, its stConn field SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].
   4. If a file in [MS-ODCFF] is referenced by the workbook file:
      1. SourceFile element of OfficeDataConnection ([MS-ODCFF] section 2.7.1.1) MUST NOT be present.
      2. Parameter elements of CT_Connection ([MS-ODCFF] section 2.2.1) MUST be ignored.

2. If the language represented by uiCultureName is unsupported on the protocol server, the protocol server MAY<20> fall back to some other supported language. uiCultureName MAY<21> be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

3. If the language represented by dataCultureName is unsupported on the protocol server, the protocol server MAY<22> fall back to some other supported language. dataCultureName MAY<23> be empty, in which case the determination of which language to default to is protocol-server-implementation specific.
4. If the operation succeeds, the protocol server MUST return the session identifier of the newly created session (2) in the response message, to be used in any future operations in the newly created session (2).

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileOpenNotFound</td>
</tr>
<tr>
<td>FileOpenAccessDenied</td>
</tr>
<tr>
<td>FileCorrupt</td>
</tr>
<tr>
<td>GenericFileOpenError</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbookForEditingSoapIn</td>
<td>The request WSDL message for the OpenWorkbookForEditing WSDL operation.</td>
</tr>
<tr>
<td>OpenWorkbookForEditingSoapOut</td>
<td>The response WSDL message for the OpenWorkbookForEditing WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.21.1.1 OpenWorkbookForEditingSoapIn
The request WSDL message for the OpenWorkbookForEditing WSDL operation.
The SOAP action value is:

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

The SOAP body contains the OpenWorkbookForEditing element.

3.1.4.21.1.2 OpenWorkbookForEditingSoapOut
The response WSDL message for the OpenWorkbookForEditing WSDL operation.
The SOAP body contains the OpenWorkbookForEditingResponse element.

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenWorkbookForEditing</td>
<td>The input data for the OpenWorkbookForEditing WSDL operation.</td>
</tr>
<tr>
<td>OpenWorkbookForEditingResponse</td>
<td>The result data for the OpenWorkbookForEditing WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.21.2.1 OpenWorkbookForEditing

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

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

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

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

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

### 3.1.4.21.2.2 OpenWorkbookForEditingResponse

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

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

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

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.21.3 Complex Types

None.
3.1.4.21.4 Simple Types
None.

3.1.4.21.5 Attributes
None.

3.1.4.21.6 Groups
None.

3.1.4.21.7 Attribute Groups
None.

3.1.4.22 Refresh

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

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

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

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

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

3.1.4.22.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshSoapIn</td>
<td>The request WSDL message for the Refresh WSDL operation.</td>
</tr>
<tr>
<td>RefreshSoapOut</td>
<td>The response WSDL message for the Refresh WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.22.1.1 RefreshSoapIn

The request WSDL message for the Refresh WSDL operation.

The SOAP action value is:
The SOAP body contains the Refresh element.

3.1.4.22.1.2 RefreshSoapOut

The response WSDL message for the Refresh WSDL operation.

The SOAP body contains the RefreshResponse element.

3.1.4.22.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>The input data for the Refresh WSDL operation.</td>
</tr>
<tr>
<td>RefreshResponse</td>
<td>The result data for the Refresh WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.22.2.1 Refresh

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

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

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

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

3.1.4.22.2.2 RefreshResponse

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

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

**status:** An array of Status elements as specified in section 2.2.4.2.
3.1.4.22.3 Complex Types
None.

3.1.4.22.4 Simple Types
None.

3.1.4.22.5 Attributes
None.

3.1.4.22.6 Groups
None.

3.1.4.22.7 Attribute Groups
None.

3.1.4.23 RefreshDetached
MUST NOT be used.

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

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

The RefreshDetachedSoapIn and RefreshDetachedSoapOut messages MUST NOT be used.

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshDetachedSoapIn</td>
<td>The request WSDL message for the RefreshDetached WSDL operation.</td>
</tr>
<tr>
<td>RefreshDetachedSoapOut</td>
<td>The response WSDL message for the RefreshDetached WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.23.1.1 RefreshDetachedSoapIn
The request WSDL message for the RefreshDetached WSDL operation.
The SOAP action value is:

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

The SOAP body contains the RefreshDetached element.
3.1.4.23.1.2 RefreshDetachedSoapOut

The response WSDL message for the RefreshDetached WSDL operation.

The SOAP body contains the RefreshDetachedResponse element.

3.1.4.23.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshDetached</td>
<td>The input data for the RefreshDetached WSDL operation.</td>
</tr>
<tr>
<td>RefreshDetachedResponse</td>
<td>The result data for the RefreshDetached WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.23.2.1 RefreshDetached

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

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

sessionId: MUST NOT be used.
connectionDetails: MUST NOT be used.
requestId: MUST NOT be used.

3.1.4.23.2.2 RefreshDetachedResponse

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

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

requestId: MUST NOT be used.
status: MUST NOT be used.
**detailedRefreshResult:** MUST NOT be used.

### 3.1.4.23.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfConnectionDetails</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ArrayOfDetailedRefreshResult</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>ConnectionDetails</td>
<td>MUST NOT be used.</td>
</tr>
<tr>
<td>DetailedRefreshResult</td>
<td>MUST NOT be used.</td>
</tr>
</tbody>
</table>

#### 3.1.4.23.3.1 ArrayOfConnectionDetails

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

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

**ConnectionDetails:** MUST NOT be used.

#### 3.1.4.23.3.2 ArrayOfDetailedRefreshResult

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

MUST NOT be used.

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

**DetailedRefreshResult:** MUST NOT be used.

#### 3.1.4.23.3.3 ConnectionDetails

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

MUST NOT be used.

```xml
<xs:complexType name="ConnectionDetails" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="ConnectionName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="TargetApplicationId" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```
ConnectionName: MUST NOT be used.

TargetApplicationId: MUST NOT be used.

3.1.4.23.3.4 DetailedRefreshResult

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

MUST NOT be used.

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

ConnectionName: MUST NOT be used.

ErrorMessage: MUST NOT be used.

3.1.4.23.4 Simple Types

None.

3.1.4.23.5 Attributes

None.

3.1.4.23.6 Groups

None.

3.1.4.23.7 Attribute Groups

None.

3.1.4.24 RefreshEx

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

The following is the WSDL port type specification of the RefreshEx WSDL operation.

<wsdl:operation name="RefreshEx" xmlns:wsdl="http://schemas.xmlsoap.org/wsls/">
  <wsdl:input message="tns:RefreshExSoapIn"/>
  <wsdl:output message="tns:RefreshExSoapOut"/>
</wsdl:operation>

For details, see the Refresh operation (section 3.1.4.22).

3.1.4.24.1 Messages

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

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

The **SOAP action** value is:

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

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

### 3.1.4.24.1.2 RefreshExSoapOut

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

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

### 3.1.4.24.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshEx</td>
<td>The input data for the <strong>RefreshEx</strong> <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>RefreshExResponse</td>
<td>The result data for the <strong>RefreshEx</strong> <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.24.2.1 RefreshEx

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

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

**sessionId**: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operation, but MAY instead be obtained in any other way.
connectionName: A name of a data connection in the workbook. MUST NOT be longer than 255 characters.


3.1.4.24.2.2 RefreshExResponse

The RefreshExResponse element specifies the result data for the RefreshEx WSDL operation.

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

status: An array of Status elements as specified in section 2.2.4.2.

3.1.4.24.3 Complex Types

None.

3.1.4.24.4 Simple Types

None.

3.1.4.24.5 Attributes

None.

3.1.4.24.6 Groups

None.

3.1.4.24.7 Attribute Groups

None.

3.1.4.25 SaveWorkbook

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

The following is the WSDL port type specification of the SaveWorkbook WSDL operation.

```xml
<wsdl:operation name="SaveWorkbook" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SaveWorkbookSoapIn"/>
  <wsdl:output message="tns:SaveWorkbookSoapOut"/>
</wsdl:operation>
```

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

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileSaveGeneric</td>
</tr>
<tr>
<td>OperationNotSupportedInViewOnly</td>
</tr>
<tr>
<td>SaveFailed</td>
</tr>
<tr>
<td>SavingUnsupportedInViewMode</td>
</tr>
<tr>
<td>SavingWorkbookAccessDenied</td>
</tr>
<tr>
<td>SavingWorkbookLockedOrCheckedOut</td>
</tr>
<tr>
<td>SavingWorkbookNoWorkbookName</td>
</tr>
</tbody>
</table>

3.1.4.25.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbookSoapIn</td>
<td>The request WSDL message for the SaveWorkbook WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookSoapOut</td>
<td>The response WSDL message for the SaveWorkbook WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.25.1.1 SaveWorkbookSoapIn

The request WSDL message for the SaveWorkbook WSDL operation.

The SOAP action value is:

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

The SOAP body contains the SaveWorkbook element.

3.1.4.25.1.2 SaveWorkbookSoapOut

The response WSDL message for the SaveWorkbook WSDL operation.

The SOAP body contains the SaveWorkbookResponse element.

3.1.4.25.2 Elements

The following table summarizes the XML schema element definitions that are specific to this operation.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbook</td>
<td>The input data for the <strong>SaveWorkbook</strong> WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookResponse</td>
<td>The result data for the <strong>SaveWorkbook</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.25.2.1 SaveWorkbook

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

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

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

### 3.1.4.25.2.2 SaveWorkbookResponse

The **SaveWorkbookResponse** element specifies the result data for the **SaveWorkbook** WSDL operation.

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

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.25.3 Complex Types

None.

### 3.1.4.25.4 Simple Types

None.

### 3.1.4.25.5 Attributes

None.

### 3.1.4.25.6 Groups

None.

### 3.1.4.25.7 Attribute Groups

None.
None.

3.1.4.26  SaveWorkbookCopy

This operation is used to save a workbook associated with an open session to a specified location, and in a specified format.

The following is the WSDL port type specification of the SaveWorkbookCopy WSDL operation.

```xml
<wsdl:operation name="SaveWorkbookCopy" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SaveWorkbookCopySoapIn"/>
  <wsdl:output message="tns:SaveWorkbookCopySoapOut"/>
</wsdl:operation>
```

The protocol client sends a SaveWorkbookSoapIn request message and the protocol server responds with a SaveWorkbookSoapOut response message, as follows:

- If the protocol server cannot save the workbook in the form specified by workbookType, the protocol server returns a SOAP fault with subcode SaveFailed.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>EcsSnapshotAccessDenied</td>
</tr>
<tr>
<td>FileSaveGeneric</td>
</tr>
<tr>
<td>SaveFailed</td>
</tr>
<tr>
<td>SavingWorkbookAccessDenied</td>
</tr>
<tr>
<td>SavingWorkbookFileNameExists</td>
</tr>
<tr>
<td>SavingWorkbookInvalidFileName</td>
</tr>
<tr>
<td>SavingWorkbookInvalidLocation</td>
</tr>
</tbody>
</table>

3.1.4.26.1  Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbookCopySoapIn</td>
<td>The request WSDL message for the SaveWorkbookCopy WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookCopySoapOut</td>
<td>The response WSDL message for the SaveWorkbookCopy WSDL operation.</td>
</tr>
</tbody>
</table>
3.1.4.26.1.1 SaveWorkbookCopySoapIn
The request WSDL message for the SaveWorkbookCopy WSDL operation.
The SOAP action value is:

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

The SOAP body contains the SaveWorkbookCopy element.

3.1.4.26.1.2 SaveWorkbookCopySoapOut
The response WSDL message for the SaveWorkbookCopy WSDL operation.
The SOAP body contains the SaveWorkbookCopyResponse element.

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbookCopy</td>
<td>The input data for the SaveWorkbookCopy WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookCopyResponse</td>
<td>The result data for the SaveWorkbookCopy WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.26.2.1 SaveWorkbookCopy
The SaveWorkbookCopy element specifies the input data for the SaveWorkbookCopy WSDL operation.

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

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

workbookPath: The location of the workbook file to which to write.

- MUST be present.

If the specified value is not a Uniform Resource Identifier (URI) as specified in [RFC2396], or a Universal Naming Convention (UNC), the protocol server MUST return a SOAP fault response with subcode SavingWorkbookInvalidFileName.
The protocol server ignores the number sign ("#") and fragment, as specified in [RFC3986].

**workbookType**: A **WorkbookType** as specified in section 2.2.5.5.

**saveOptions**: A combination of 0 or more values of **SaveOptions** (section 2.2.5.3). MUST be present.

### 3.1.4.26.2.2 SaveWorkbookCopyResponse

The **SaveWorkbookCopyResponse** element specifies the result data for the **SaveWorkbookCopy WSDL operation**.

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

**status**: An array of status information as specified in section 2.2.4.2.

### 3.1.4.26.3 Complex Types

None.

### 3.1.4.26.4 Simple Types

None.

### 3.1.4.26.5 Attributes

None.

### 3.1.4.26.6 Groups

None.

### 3.1.4.26.7 Attribute Groups

None.

### 3.1.4.27 SaveWorkbookCopyDetached

MUST NOT be used.

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

```xml
<wsdl:operation name="SaveWorkbookCopyDetached" xmlns:wsdl="http://schemas.xmlsoap.org/wsd//">
  <wsdl:input message="tns:SaveWorkbookCopyDetachedSoapIn"/>
  <wsdl:output message="tns:SaveWorkbookCopyDetachedSoapOut"/>
</wsdl:operation>
```

The **SaveWorkbookCopyDetachedSoapIn** and **SaveWorkbookCopyDetachedSoapOut** messages MUST NOT be used.
3.1.4.27.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbookCopyDetachedSoapIn</td>
<td>The request WSDL message for the SaveWorkbookCopyDetached WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookCopyDetachedSoapOut</td>
<td>The response WSDL message for the SaveWorkbookCopyDetached WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.27.1.1 SaveWorkbookCopyDetachedSoapIn

The request WSDL message for the SaveWorkbookCopyDetached WSDL operation.

The SOAP action value is:

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

The SOAP body contains the SaveWorkbookCopyDetached element.

3.1.4.27.1.2 SaveWorkbookCopyDetachedSoapOut

The response WSDL message for the SaveWorkbookCopyDetached WSDL operation.

The SOAP body contains the SaveWorkbookCopyDetachedResponse element.

3.1.4.27.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaveWorkbookCopyDetached</td>
<td>The input data for the SaveWorkbookCopyDetached WSDL operation.</td>
</tr>
<tr>
<td>SaveWorkbookCopyDetachedResponse</td>
<td>The result data for the SaveWorkbookCopyDetached WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.27.2.1 SaveWorkbookCopyDetached

The SaveWorkbookCopyDetached element specifies the input data for the SaveWorkbookCopyDetached WSDL operation.

```xml
geroot element name="SaveWorkbookCopyDetached" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
        <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"/>
        <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```
<xs:element name="SaveWorkbookCopyDetachedResponse" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

**requestId:** MUST NOT be used.

**status:** MUST NOT be used.

### 3.1.4.27.3 Complex Types
None.

### 3.1.4.27.4 Simple Types
None.

### 3.1.4.27.5 Attributes
None.

### 3.1.4.27.6 Groups
None.

### 3.1.4.27.7 Attribute Groups
None.

### 3.1.4.28 SetCalculationOptions
This operation is used to set the **workbook calculation mode**.

The following is the WSDL port type specification of the **SetCalculationOptions** WSDL operation.
The protocol client sends a **SetCalculationOptionsSoapIn** request message and the protocol server responds with a **SetCalculationOptionsSoapOut** response message as follows:

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

### 3.1.4.28.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCalculationOptionsSoapIn</td>
<td>The request WSDL message for the <strong>SetCalculationOptions</strong> WSDL operation.</td>
</tr>
<tr>
<td>SetCalculationOptionsSoapOut</td>
<td>The response WSDL message for the <strong>SetCalculationOptions</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.28.1.1 SetCalculationOptionsSoapIn

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

The **SOAP action** value is:

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

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

### 3.1.4.28.1.2 SetCalculationOptionsSoapOut

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

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

### 3.1.4.28.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCalculationOptions</td>
<td>The input data for the <strong>SetCalculationOptions</strong> <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>SetCalculationOptionsResponse</td>
<td>The result data for the <strong>SetCalculationOptions</strong> <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>
3.1.4.28.2.1 SetCalculationOptions

The SetCalculationOptions element specifies the input data for the SetCalculationOptions WSDL operation.

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

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

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

3.1.4.28.2.2 SetCalculationOptionsResponse

The SetCalculationOptionsResponse element specifies the result data for the SetCalculationOptions WSDL operation.

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

**status:** An array of status information as specified in section 2.2.4.2.

3.1.4.28.3 Complex Types

None.

3.1.4.28.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkbookCalculation</td>
<td>The WorkbookCalculation type is an enumeration of a set of options that specify how the protocol server calculates the workbook.</td>
</tr>
</tbody>
</table>

3.1.4.28.4.1 WorkbookCalculation

**Namespace:** http://schemas.microsoft.com/office/excel/server/webservices
The **WorkbookCalculation** type is an enumeration of a set of options that specify how the protocol server calculates the workbook. This type is defined as follows.

```xml
<xs:simpleType name="WorkbookCalculation" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Manual"/>
        <xs:enumeration value="Automatic"/>
        <xs:enumeration value="Semiautomatic"/>
    </xs:restriction>
</xs:simpleType>
```

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>Specifies workbook is in <strong>manual calculation mode</strong>.</td>
</tr>
<tr>
<td>Automatic</td>
<td>Specifies workbook is in <strong>automatic calculation mode</strong>.</td>
</tr>
<tr>
<td>Semiautomatic</td>
<td>Specifies workbook is in automatic calculation mode, except for data tables.</td>
</tr>
</tbody>
</table>

### 3.1.4.28.5 Attributes

None.

### 3.1.4.28.6 Groups

None.

### 3.1.4.28.7 Attribute Groups

None.

### 3.1.4.29 SetCell

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

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

```xml
<wsdl:operation name="SetCell" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:input message="tns:SetCellSoapIn"/>
    <wsdl:output message="tns:SetCellSoapOut"/>
</wsdl:operation>
```

1. The **row** and **column** fields on the sheets (2) specified by **sheetName** specify the location of the worksheet cell to set.

2. **cellValue** MUST either be empty and have the **xsi:nil** attribute set to **true** or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string**, or **unsigned byte**.
   1. If the value is of type **string**, it MUST NOT be longer than 32,767 characters.
   2. If the protocol server does not support the value it MUST return a **SOAP fault**.
3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormulaParseFailed</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>CannotEditPivotTable</td>
</tr>
</tbody>
</table>

### 3.1.4.29.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCellSoapIn</td>
<td>The request WSDL message for the SetCell <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>SetCellSoapOut</td>
<td>The response WSDL message for the SetCell <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.29.1.1 SetCellSoapIn

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

The **SOAP action** value is:

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

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

#### 3.1.4.29.1.2 SetCellSoapOut

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

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

### 3.1.4.29.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCell</td>
<td>The input data for the SetCell <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>
### 3.1.4.29.2.1 SetCell

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

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

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

**sheetName:** The name of the **sheet (2)** that contains the **worksheet cell** to set. See section 2.2.5.4. MUST be present. MUST NOT be empty.

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

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

**cellValue:** The value to set in the worksheet cell.

### 3.1.4.29.2.2 SetCellResponse

The **SetCellResponse** element specifies the result data for the **SetCell WSDL operation**.

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

**status:** An array of status information as specified in section 2.2.4.

### 3.1.4.29.3 Complex Types

None.

### 3.1.4.29.4 Simple Types

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCellResponse</td>
<td>The result data for the <strong>SetCell</strong> WSDL operation.</td>
</tr>
</tbody>
</table>
None.

3.1.4.29.5 Attributes
None.

3.1.4.29.6 Groups
None.

3.1.4.29.7 Attribute Groups
None.

3.1.4.30 SetCellA1
This operation is used to set a value to a specific worksheet cell in the workbook.
The following is the WSDL port type specification of the SetCellA1 WSDL operation.

```xml
<wsdl:operation name="SetCellA1" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:input message="tns:SetCellA1SoapIn"/>
<wsdl:output message="tns:SetCellA1SoapOut"/>
</wsdl:operation>
```

The protocol client sends a SetCellA1SoapIn request message, and the protocol server responds with a SetCellA1SoapOut response message, as follows:

1. The sheet (2) that contains the worksheet cell that will be set is determined in the following manner:
   - If rangeName specifies a sheet (2), that sheet (2) MUST be used.
   - Otherwise, the sheet (2) specified in sheetName MUST be used, as specified in section 2.2.5.4.
   - If neither sheetName nor rangeName specify a sheet (2), the protocol server MAY<24> use any sheet (2) in the workbook.
2. If sheetName is empty and rangeName specifies a sheet (2), as defined in section 2.2.5.2, the protocol server MUST use the sheet (2) that is specified by rangeName to locate the range.
3. cellValue MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: boolean, dateTime, decimal, double, float, int, short, string, or unsigned byte.
   1. If the value is of type string, it MUST NOT be longer than 32767 characters.
   2. If the protocol server does not support the value, it MUST return a SOAP fault.
   3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.
4. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section 2.2.9.2, but MAY instead contain any subcode specified in section 2.2.9.1 or any other subcode.
### 3.1.4.30.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCellA1SoapIn</td>
<td>The request WSDL message for the <em>SetCellA1</em> WSDL operation.</td>
</tr>
<tr>
<td>SetCellA1SoapOut</td>
<td>The response WSDL message for the <em>SetCellA1</em> WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.30.1.1 SetCellA1SoapIn

The request **WSDL message** for the *SetCellA1* **WSDL operation**.

The **SOAP action** value is:

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

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

#### 3.1.4.30.1.2 SetCellA1SoapOut

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

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

### 3.1.4.30.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCellA1</td>
<td>The input data for the <em>SetCellA1</em> <strong>WSDL operation</strong>.</td>
</tr>
<tr>
<td>SetCellA1Response</td>
<td>The result data for the <em>SetCellA1</em> <strong>WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

---

[MS-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019
3.1.4.30.2.1 SetCellA1

The SetCellA1 element specifies the input data for the SetCellA1 WSDL operation.

```
<xs:element name="SetCellA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

**sheetName:** The name of the sheet (2) that contains the worksheet cell to set. See section 2.2.5.4. MUST be present. MUST NOT be empty.

**rangeName:** The range of the worksheet cell to set. See section 2.2.5.2. MUST be present.

**cellValue:** The value to set in the worksheet cell.

3.1.4.30.2.2 SetCellA1Response

The SetCellA1Response element specifies the result data for the SetCellA1 WSDL operation.

```
<xs:element name="SetCellA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**status:** An array of status information as specified in section 2.2.4.2.

3.1.4.30.3 Complex Types

None.

3.1.4.30.4 Simple Types

None.

3.1.4.30.5 Attributes

None.

3.1.4.30.6 Groups

None.

3.1.4.30.7 Attribute Groups
None.

### 3.1.4.31 SetParameters

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

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

```xml
<wsdl:operation name="SetParameters" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetParametersSoapIn"/>
  <wsdl:output message="tns:SetParametersSoapOut"/>
</wsdl:operation>
```

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

- For each **ParameterInfo** (section 3.1.4.31.3.2) in **parameters**, if the workbook parameter is a **PivotTable page field**:
  - If the number of elements in the **Values** array of **ParameterInfo** is equal to or greater than 1, the protocol server uses the values as the **PivotTable manual filter**.
  - If the number of element in the **Values** array of **ParameterInfo** is 0, the protocol server changes the PivotTable manual filter to the **default filter value**.

- For each **ParameterInfo** in **parameters**, if the workbook parameter is a **slicer page filter**:
  - If the number of elements in the **Values** array of **ParameterInfo** is equal to or greater than 1, the protocol server uses the values as the slicer manual filter.
  - If the number of elements in the **Values** array of **ParameterInfo** is 0, the protocol server changes the slicer manual filter to the default filter value.

- For each **ParameterInfo** in **parameters**, if the workbook parameter is neither a slicer page filter nor a PivotTable page filter:
  - If the number of elements in the **Values** array of **ParameterInfo** is not exactly 1, the protocol server ignores all elements in the array.
  - If the number of elements in the **Values** array of **ParameterInfo** is exactly 1, the protocol server applies that value to the parameter.

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

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormulaEditingNotEnabled</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>ParameterDoesNotExist</td>
</tr>
<tr>
<td>SetParameterFailure</td>
</tr>
</tbody>
</table>
3.1.4.31.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetParametersSoapIn</td>
<td>The request WSDL message for the SetParameters WSDL operation.</td>
</tr>
<tr>
<td>SetParametersSoapOut</td>
<td>The response WSDL message for the SetParameters WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.31.1.1 SetParametersSoapIn

The request WSDL message for the SetParameters WSDL operation.

The SOAP action value is:

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

The SOAP body contains the SetParameters element.

3.1.4.31.1.2 SetParametersSoapOut

The response WSDL message for the SetParameters WSDL operation.

The SOAP body contains the SetParametersResponse element.

3.1.4.31.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetParameters</td>
<td>The input data for the SetParameters WSDL operation.</td>
</tr>
<tr>
<td>SetParametersResponse</td>
<td>The result data for the SetParameters WSDL operation.</td>
</tr>
</tbody>
</table>

3.1.4.31.2.1 SetParameters

The SetParameters element specifies the input data for the SetParameters WSDL operation.

```xml
<xs:element name="SetParameters" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="parameters" type="tns:ArrayOfParameterInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
sessionId: A **session identifier**. MUST be present. SHOULD have a value that has been obtained from a previous call to the OpenWorkbook, OpenWorkbookEx, OpenWorkbookForEditing, or NewWorkbook operations, but MAY instead be obtained in any other way.

parameters: An array of **ParameterInfo** elements (section 3.1.4.31.3.2) to be set in the workbook, as specified in section 3.1.4.31.3.1. MUST be present.

### 3.1.4.31.2.2 SetParametersResponse

The **SetParametersResponse** element specifies the result data for the **SetParameters WSDL operation**.

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

**status**: An array of **Status** elements as specified in section 2.2.4.2.

### 3.1.4.31.3 Complex Types

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayOfParameterInfo</td>
<td>An array of <strong>ParameterInfo</strong> elements.</td>
</tr>
<tr>
<td>ParameterInfo</td>
<td>Represents a value or a list of values to be set into a <strong>workbook parameter</strong> in the workbook.</td>
</tr>
</tbody>
</table>

#### 3.1.4.31.3.1 ArrayOfParameterInfo

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

An array of **ParameterInfo** elements defined as follows. The number of element in the **ArrayOfParameterInfo** element MUST be less than or equal to 255.

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

**ParameterInfo**: A **workbook parameter** as specified in section 3.1.4.31.3.2.

#### 3.1.4.31.3.2 ParameterInfo

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

Represents a value or a list of values to be set into a **workbook parameter** in the workbook and is defined as follows.
Name: The name of the workbook parameter. MUST NOT exceed 255 characters.

Values: An array of string values to set in the workbook parameter as specified in section 2.2.4.3.

3.1.4.31.4 Simple Types
None.

3.1.4.31.5 Attributes
None.

3.1.4.31.6 Groups
None.

3.1.4.31.7 Attribute Groups
None.

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

The following is the WSDL port type specification of the SetRange WSDL operation.

<wsdl:operation name="SetRange" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SetRangeSoapIn"/>
  <wsdl:output message="tns:SetRangeSoapOut"/>
</wsdl:operation>

1. The rangeCoordinates on the sheet (2) specified by sheetName; specify the location of the worksheet cells to set.
2. If the user does not have the Open Item permission on the workbook, rangeValues SHOULD <25> contain only one item.
3. Each element inside the rangeValues element MUST be of type ArrayOfAnyType and be a row of worksheet cells. The number of rows in rangeValues MUST be equal to the number of rows specified in the rangeCoordinates element.
4. Each element inside each of the ArrayOfAnyType type nested inside rangeValues MUST be a single worksheet cell. The number of elements inside each such nested ArrayOfAnyType MUST be equal to the number of columns specified in the rangeCoordinates element.
5. All worksheet cells in the rangeValues element MUST either be empty and have the xsi:nil attribute set to true or one be of the following types: boolean, dateTime, decimal, double, float, int, short, string, or unsigned byte.
   1. If the value is of type string, it MUST NOT be longer than 32,767 characters.
   2. If the protocol server does not support the value, it MUST return a SOAP fault.
   3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.
6. If there was an application error on the protocol server during the operation, the protocol server MUST return a SOAP fault response, which SHOULD contain one of the following subcodes or any other subcodes specified in section 2.2.9.2, but MAY instead contain any subcode specified in section 2.2.9.1 or any other subcode.

<table>
<thead>
<tr>
<th>Subcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormulaParseFailed</td>
</tr>
<tr>
<td>InvalidSheetName</td>
</tr>
<tr>
<td>DimensionAndArrayMismatch</td>
</tr>
<tr>
<td>SheetRangeMismatch</td>
</tr>
<tr>
<td>RangeParseError</td>
</tr>
</tbody>
</table>

### 3.1.4.32.1 Messages

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

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetRangeSoapIn</td>
<td>The request WSDL message for the SetRange WSDL operation.</td>
</tr>
<tr>
<td>SetRangeSoapOut</td>
<td>The response WSDL message for the SetRange WSDL operation.</td>
</tr>
</tbody>
</table>

#### 3.1.4.32.1.1 SetRangeSoapIn

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

The **SOAP action** value is:

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

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

#### 3.1.4.32.1.2 SetRangeSoapOut

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

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

### 3.1.4.32.2 Elements

The following table summarizes the **XML schema** element definitions that are specific to this operation.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetRange</td>
<td>The input data for the SetRange WSDL operation.</td>
</tr>
<tr>
<td>SetRangeResponse</td>
<td>The result data for the SetRange WSDL operation.</td>
</tr>
</tbody>
</table>

### 3.1.4.32.2.1 SetRange

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

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

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

**sheetName**: The name of the worksheet that contains the **range** to set. See section 2.2.5.4. MUST be present. MUST NOT be empty.

**rangeCoordinates**: The range to set. See section 2.2.5.2. MUST be present. MUST NOT be empty.

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

### 3.1.4.32.2.2 SetRangeResponse

The **SetRangeResponse** element specifies the result data for the **SetRange** WSDL operation.

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

**status**: An array of status information as specified in section 2.2.4.2.

### 3.1.4.32.3 Complex Types

None.

### 3.1.4.32.4 Simple Types

None.
### 3.1.4.32.5 Attributes

None.

### 3.1.4.32.6 Groups

None.

### 3.1.4.32.7 Attribute Groups

None.

### 3.1.4.33 SetRangeA1

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

The following is the WSDL port type specification of the SetRangeA1 WSDL operation.

```xml
<wSDL:operation name="SetRangeA1" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
  <wSDL:input message="tns:SetRangeA1SoapIn"/>
  <wSDL:output message="tns:SetRangeA1SoapOut"/>
</wSDL:operation>
```

1. The sheet (2) that contains the range that will be set is determined in the following manner:
   - If `rangeName` specifies a sheet (2), that sheet (2) MUST be used.
   - Otherwise, the sheet (2) specified in `sheetName` MUST be used, as specified in section 2.2.5.4.
     - If neither `sheetName` nor `rangeName` specify a sheet (2), the protocol server MAY use any sheet (2) in the workbook.
2. If `sheetName` is empty and `rangeName` specifies a sheet (2), as defined in section 2.2.5.2, the sheet (2) that is specified by `rangeName` MUST be used to locate the range.
3. Each element inside the `rangeValues` element MUST be of type `ArrayOfAnyType` and be a row of worksheet cells. The number of rows in `rangeValues` MUST be equal to the number of rows specified in the `rangeName` element.
4. Each element inside each of the `ArrayOfAnyType` type nested inside `rangeValues` MUST be a single worksheet cell. The number of elements inside each such nested `ArrayOfAnyType` MUST be equal to the number of columns specified in the `rangeName` element.
5. All worksheet cells in the `rangeValues` element MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: boolean, dateTime, decimal, double, float, int, short, string, or unsigned byte.
   - If the value is of type `string`, it MUST NOT be longer than 32767 characters.
   - If the protocol server does not support the value, it MUST return a SOAP fault.
   - If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.
6. If the user does not have the Open Item permission on the workbook, `rangeValues` SHOULD contain only one worksheet cell.

### 3.1.4.33.1 Messages

The following table summarizes the set of WSDL message definitions that are specific to this operation.
### 3.1.4.33.1.1 SetRangeA1SoapIn

The request **WSDL message** for the **SetRangeA1 WSDL operation**.

The **SOAP action** value is:


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

### 3.1.4.33.1.2 SetRangeA1SoapOut

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

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

### 3.1.4.33.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetRangeA1</td>
<td>The input data for the <strong>SetRangeA1 WSDL operation</strong>.</td>
</tr>
<tr>
<td>SetRangeA1Response</td>
<td>The result data for the <strong>SetRangeA1 WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

#### 3.1.4.33.2.1 SetRangeA1

The **SetRangeA1** element specifies the input data for the **SetRangeA1 WSDL operation**.

```xml
<xs:element name="SetRangeA1" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeValues" type="tns:ArrayOfAnyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**sessionId**: A session identifier (2). MUST be present in the request message. SHOULD have a value that has been obtained from a previous call to the **OpenWorkbook**, **OpenWorkbookEx**, **OpenWorkbookForEditing**, or **NewWorkbook** operations, but MAY instead be obtained in any other way.
**sheetName:** The name of the **sheet (2)** that contains the **range** to set. See section 2.2.5.4. MUST be present.

**rangeName:** The name of the range to set. See section 2.2.5.2. MUST be present.

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

### 3.1.4.33.2.2 SetRangeA1Response

The **SetRangeA1Response** element specifies the result data for the **SetRangeA1 WSDL operation**.

```xml
<xs:element name="SetRangeA1Response" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**status:** An array of status information as specified in section 2.2.4.2.

### 3.1.4.33.3 Complex Types

None.

### 3.1.4.33.4 Simple Types

None.

### 3.1.4.33.5 Attributes

None.

### 3.1.4.33.6 Groups

None.

### 3.1.4.33.7 Attribute Groups

None.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.
4 Protocol Examples

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

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

4.1 Getting the Value of a Worksheet Cell

This example shows a simple, and typical, flow of how this protocol is used to enable a protocol client to get calculated results from a workbook. This example assumes a workbook has already been created and is stored in a location that the protocol server can load workbook files from.

In this example, a workbook is loaded on the protocol server, recalculated, a value is fetched, and the workbook is closed. The following UML diagram illustrates this example.

![UML Diagram](image)

Figure 3: Getting the value of a worksheet cell example

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

1. Load the workbook.

Request

```xml
<?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">
      <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"
xmni:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<OpenWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
<OpenWorkbookResult/>
</OpenWorkbookResponse>
</soap:Body>
</soap:Envelope>

2. Recalculate the workbook.

Request

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmni: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.V23.5791QXInMoJutkHzkZDd90.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#0000#0000-03-00-02T02:00:00:0000=#-06036.4d387825-9c48-4d48-98fe-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"
xmni:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<CalculateWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
</soap:Body>
</soap:Envelope>

3. Fetch the specific value from a worksheet cell within the workbook.

Request

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmni:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
</soap:Body>
</soap:Envelope>
<soap:Body>
  <GetCellA1
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
    <sessionId>36.c22a23f3-976c-4933-aa85-74c6f06c51b5164.1-V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-7894661941631.N</sessionId>
    <sheetName>Sheet1</sheetName>
    <rangeName>A1</rangeName>
    <formatted>false</formatted>
  </GetCellA1>
</soap:Body>
</soap:Envelope>

Response

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetCellA1Response
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <GetCellA1Result xsi:type="xsd:double">5</GetCellA1Result>
    </GetCellA1Response>
  </soap:Body>
</soap:Envelope>

4. Close the workbook session on the protocol server.

Request

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbook
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-74c6f06c51b5164.1-V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-7894661941631.N</sessionId>
    </CloseWorkbook>
  </soap:Body>
</soap:Envelope>

Response

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
4.2 Refreshing Data and Retrieving a Workbook

This example illustrates how this protocol is used to enable a more complex series of actions, including refreshing connected data and retrieving the resulting workbook after a calculation. This example assumes a workbook has already been created and is stored in a location that the workbook protocol server can load files from. In this example a workbook is loaded on the protocol server, some values are set into the workbook, the connected data is refreshed, the workbook is recalculated, and finally the desired result of that calculation is retrieved and the workbook is closed. The following UML diagram illustrates this example.

![UML diagram](image)

**Figure 4: Refresh data and retrieve workbook example**

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

1. Load the workbook.

**Request**

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <OpenWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices"/>
  </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-aa85-74cf60c51b5164.1.V23.494r81aHjSSiziqYhY1U2B590.5.en-US5.en-US73.+04800000-11-00-01T02:00:00:00000#00000-03-00-02T02:00:00:00000#006036.4d387825-9c48-4d48-98fe-7894661941631.N</OpenWorkbookResult>
    </OpenWorkbookResponse>
  </soap:Body>
</soap:Envelope>

2. Set the value of the worksheet cell.

Request

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetCellA1 xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-74cf60c51b5164.1.V23.494r81aHjSSiziqYhY1U2B590.5.en-US5.en-US73.+04800000-11-00-01T02:00:00:00000#00000-03-00-02T02:00:00:00000#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>
  </soap:Body>
</soap:Envelope>
3. Refresh the connected data in the workbook.

Request

```xml
<?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.V23.494r81aHjSSiziqJhY1U2B590.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#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
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RefreshResponse xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

4. Recalculate the workbook.

Request

```xml
<?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.V23.494r81aHjSSiziqJhY1U2B590.5.en-US5.en-US73.+0480#0000-11-00-01T02:00:00:0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-7894661941631.N</sessionId>
      <calculateType>CalculateFull</calculateType>
    </CalculateWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response
5. Fetch the workbook byte stream.

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>
    <GetWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-74c6f06c51b5164.1.V23.494r81aHjSSiziqThY1U2B590.5.en-UsU5.en-Us73.0+480#0000-11-00-01T02:00:00.0000000+#0000-03-00-02T02:00:00.0000000-006036.4d387825-9c48-4d48-98fe-789466194163.1.N</sessionId>
      <workbookType>FullWorkbook</workbookType>
    </GetWorkbook>
  </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>
    <GetWorkbookResponse xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <GetWorkbookResult>UEsDBBBQABgAAIAAAAIQAezCq/7QEAAAYKAAAAAgCWUNVbmr1bwnRlfV1w6XNdLht
bCC1BAIooAACAAAAAAAACACAAAAAABCAAAAAAAACACAAAAAABCAAAAAAACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAAACACAAAAAABCAAAAAA/

  </GetWorkbookResponse>
</soap:Body>
</soap:Envelope>
6. 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/">
  <soap:Body>
    <CloseWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-74c6f06c51b5164.1.V23.494r81aHjSSiziqJhY1U2B590.5.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/">
  <soap:Body>
  </soap:Body>
</soap:Envelope>
5 Security

5.1 Security Considerations for Implementers

There are security risks associated with exposing session identifiers. If the session identifier is exposed, it is possible for an attacker to read information from, or modify data in, a session on the protocol server. An implementer of this protocol is to consider keeping session identifiers protected. There could be some cases where it is desirable to expose the session identifier, but implementers are to use caution in how they expose session identifiers and consider the security risks.

5.2 Index of Security Parameters

None.
6  Appendix A: Full WSDL

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/
 xmlns:http="http://schemas.xmlsoap.org/wsdl/http/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://schemas.microsoft.com/office/excel/server/webservices"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
 targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
     targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices">
      <xs:element name="GetApiVersion">
        <xs:complexType/>
      </xs:element>
      <xs:element name="GetApiVersionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetApiVersionResult" type="xs:string"/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <xs:complexType name="ArrayOfStatus">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Status" nillable="true" type="tns:Status"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="Status">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Severity" type="xs:string"/>
        <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="NewWorkbook">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="path" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="NewWorkbookResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="NewWorkbookResult" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="OpenWorkbookForEditing">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
          <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs: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>
  </wsdl:types>
</wsdl:definitions>
```
<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:complexType name="WorkbookType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FullWorkbook"/>
    <xs:enumeration value="FullSnapshot"/>
    <xs:enumeration value="PublishedItemsSnapshot"/>
  </xs:restriction>
</xs:complexType>

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

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

<xs:element name="SaveWorkbookCopyDetached">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"/>
      <xs:element minOccurs="0" maxOccurs="1" name="requestId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

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

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

<xs:complexType name="RangeCoordinates">
  <xs:sequence>
  </xs:sequence>
</xs:complexType>
<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>
</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>
</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>
</xs:complexType>
</xs:element>
</xs:complexType>
<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:complexType name="ArrayOfAnyType">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="anyType" nillable="true"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</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: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 minOccurs="0" maxOccurs="1" name="GetWorkbookResult">
  <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
</xs:element>
</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" nillable="true" maxOccurs="unbounded" name="ParameterInfo" nillable="true" type="tns:ParameterInfo"/>
    </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" nillable="true"/>
      <xs:element minOccurs="0" maxOccurs="1" name="chartName" type="xs:string" nillable="true" maxOccurs="unbounded" name="string" nillable="true" 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:complexType name="WorkbookModelInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Server" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Database" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="SessionInteractivityTimeout" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Version" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

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

<xs:element name="EnsureWorkbookModel">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

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

<xs:element name="OpenWorkbookEx">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="exclusive" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

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

<xs:element name="GetWorkbookConnections">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType>
  <xs:element name="RangeReference">
    <xs:restriction base="xs:string"/>
  </xs:element>
  <xs:simpleType name="SheetName">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

<wsdl:portType name="ExcelServiceSoap">
  <wsdl:operation name="GetApiVersion">
    <wsdl:input message="tns:GetApiVersionSoapIn"/>
    <wsdl:output message="tns:GetApiVersionSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="NewWorkbook">
    <wsdl:input message="tns:NewWorkbookSoapIn"/>
    <wsdl:output message="tns:NewWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="OpenWorkbookForEditing">
    <wsdl:input message="tns:OpenWorkbookForEditingSoapIn"/>
    <wsdl:output message="tns:OpenWorkbookForEditingSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="OpenWorkbook">
    <wsdl:input message="tns:OpenWorkbookSoapIn"/>
    <wsdl:output message="tns:OpenWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CloseWorkbook">
    <wsdl:input message="tns:CloseWorkbookSoapIn"/>
    <wsdl:output message="tns:CloseWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="Refresh">
    <wsdl:input message="tns:RefreshSoapIn"/>
    <wsdl:output message="tns:RefreshSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbook">
    <wsdl:input message="tns:SaveWorkbookSoapIn"/>
    <wsdl:output message="tns:SaveWorkbookSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbookCopy">
    <wsdl:input message="tns:SaveWorkbookCopySoapIn"/>
    <wsdl:output message="tns:SaveWorkbookCopySoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SaveWorkbookCopyDetached">
    <wsdl:input message="tns:SaveWorkbookCopyDetachedSoapIn"/>
    <wsdl:output message="tns:SaveWorkbookCopyDetachedSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="Calculate">
    <wsdl:input message="tns:CalculateSoapIn"/>
    <wsdl:output message="tns:CalculateSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CalculateA1">
    <wsdl:input message="tns:CalculateA1SoapIn"/>
    <wsdl:output message="tns:CalculateA1SoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetRange">
    <wsdl:input message="tns:GetRangeSoapIn"/>
    <wsdl:output message="tns:GetRangeSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetRangeA1">
    <wsdl:input message="tns:GetRangeA1SoapIn"/>
    <wsdl:output message="tns:GetRangeA1SoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCell">
    <wsdl:input message="tns:GetCellSoapIn"/>
    <wsdl:output message="tns:GetCellSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="GetCellA1">
    <wsdl:input message="tns:GetCellA1SoapIn"/>
<!-- wsdl:operation name="RefreshDetached" -->
<wsdl:input message="tns:RefreshDetachedSoapIn"/>
<wsdl:output message="tns:RefreshDetachedSoapOut"/>
</wsdl:operation>

<!-- wsdl:operation -->
<soap:operation
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</soap: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/OpenWorkbookForEditing"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</soap: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>
</soap:operation>

<!-- wsdl:operation name="CloseWorkbook" -->
<soap:operation
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</soap: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>
</soap:operation>
<wsdl:input>
</wsdl:input>
<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>
</wsdl:input>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="SaveWorkbookCopyDetached">
<soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopyDetached"
style="document"/>
<wsdl:input>
</wsdl:input>
<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>
</wsdl:input>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="CalculateA1">
<soap:operation
style="document"/>
<wsdl:input>
</wsdl:input>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRange">
<soap:operation
style="document"/>
<wsdl:input>
</wsdl:input>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetRangeA1">
<soap:operation
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
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
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/GetPublishedItemNames"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetSheetNames">
<soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames"
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetRange">
<soap:operation
style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SetRangeA1"/>
<wsdl:input>
  <soap:body use="literal"/>
</wsdl:input>
</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/SetCalculationOptions"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetChartImageUrl">
  <soap:operation
    soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="EnsureWorkbookModel">
  <soap:operation
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OpenWorkbookEx">
  <soap:operation
    soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookEx"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetWorkbookConnections">
  <soap:operation
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:output>
  <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="RefreshEx">
  <soap:operation
    soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshEx"
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="RefreshDetached">
  <soap:operation
    style="document"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:binding name="ExcelServiceSoap12" type="tns:ExcelServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetApiVersion">
    <soap12:operation
      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/OpenWorkbookForEditing"
      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:binding>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="CloseWorkbook">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="Refresh">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SaveWorkbook">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SaveWorkbookCopy">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="SaveWorkbookCopyDetached">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>

<wsdl:operation name="Calculate">
<wsdl:input>
  <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="CalculateA1">
  <soap12:operation
  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
  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
  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
  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
  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/GetPublishedItemNames"
  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">
    style="document"/>
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="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">
        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">
          style="document"/>
          <wsdl:input>
            <soap12:body use="literal"/>
          </wsdl:input>
          <wsdl:output>
            <soap12:body use="literal"/>
          </wsdl:output>
          </wsdl:operation>
          <wsdl:operation name="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">
              soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformation"
              style="document"/>
              <wsdl:input>
                <soap12:body use="literal"/>
              </wsdl:input>
              <wsdl:output>
                <soap12:body use="literal"/>
              </wsdl:output>
              </wsdl:operation>
<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
    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/SetCalculationOptions"
    style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="GetChartImageUrl">
  <soap12:operation
    soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
    style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="EnsureWorkbookModel">
  <soap12:operation
    style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
</wsdl:operation>
<wsdl:output>
  <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="OpenWorkbookEx">
  <soap12:operation
  soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookEx"
  style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetWorkbookConnections">
  <soap12:operation
  style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RefreshEx">
  <soap12:operation
  soapAction="http://schemas.microsoft.com/office/excel/server/webservices/RefreshEx"
  style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="RefreshDetached">
  <soap12:operation
  style="document"/>
  <wsdl:input>
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap12:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:message name="CalculateA1SoapIn">
  <wsdl:part name="parameters" element="tns:CalculateA1"/>
</wsdl:message>
<wsdl:message name="CalculateA1SoapOut">
  <wsdl:part name="parameters" element="tns:CalculateA1Response"/>
</wsdl:message>
<wsdl:message name="CalculateSoapIn">
  <wsdl:part name="parameters" element="tns:Calculate"/>
</wsdl:message>
<wsdl:message name="CalculateSoapOut">
  <wsdl:part name="parameters" element="tns:CalculateResponse"/>
</wsdl:message>
<wsdl:message name="CalculateWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:CalculateWorkbook"/>
</wsdl:message>
<wsdl:message name="CalculateWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:CalculateWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="CancelRequestSoapIn">
  <wsdl:part name="parameters" element="tns:CancelRequest"/>
<wsdl:message name="CancelRequestSoapOut">
  <wsdl:part name="parameters" element="tns:CancelRequestResponse"/>
</wsdl:message>
<wsdl:message name="CloseWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:CloseWorkbook"/>
</wsdl:message>
<wsdl:message name="CloseWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:CloseWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="EnsureWorkbookModelSoapIn">
  <wsdl:part name="parameters" element="tns:EnsureWorkbookModel"/>
</wsdl:message>
<wsdl:message name="EnsureWorkbookModelSoapOut">
  <wsdl:part name="parameters" element="tns:EnsureWorkbookModelResponse"/>
</wsdl:message>
<wsdl:message name="GetApiVersionSoapIn">
  <wsdl:part name="parameters" element="tns:GetApiVersion"/>
</wsdl:message>
<wsdl:message name="GetApiVersionSoapOut">
  <wsdl:part name="parameters" element="tns:GetApiVersionResponse"/>
</wsdl:message>
<wsdl:message name="GetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:GetCellA1"/>
</wsdl:message>
<wsdl:message name="GetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:GetCellA1Response"/>
</wsdl:message>
<wsdl:message name="GetCellSoapIn">
  <wsdl:part name="parameters" element="tns:GetCell"/>
</wsdl:message>
<wsdl:message name="GetCellSoapOut">
  <wsdl:part name="parameters" element="tns:GetCellResponse"/>
</wsdl:message>
<wsdl:message name="GetChartImageUrlSoapIn">
  <wsdl:part name="parameters" element="tns:GetChartImageUrl"/>
</wsdl:message>
<wsdl:message name="GetChartImageUrlSoapOut">
  <wsdl:part name="parameters" element="tns:GetChartImageUrlResponse"/>
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNames"/>
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNamesResponse"/>
</wsdl:message>
<wsdl:message name="GetRangeA1SoapIn">
  <wsdl:part name="parameters" element="tns:GetRangeA1"/>
</wsdl:message>
<wsdl:message name="GetRangeA1SoapOut">
  <wsdl:part name="parameters" element="tns:GetRangeA1Response"/>
</wsdl:message>
<wsdl:message name="GetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:GetRange"/>
</wsdl:message>
<wsdl:message name="GetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:GetRangeResponse"/>
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapIn">
  <wsdl:part name="parameters" element="tns:GetSessionInformation"/>
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapOut">
  <wsdl:part name="parameters" element="tns:GetSessionInformationResponse"/>
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetSheetNames"/>
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetSheetNamesResponse"/>
</wsdl:message>
</wsdl:message>
<wsdl:message name="GetWorkbookConnectionsSoapIn">
  <wsdl:part name="parameters" element="tns:GetWorkbookConnections"/>
</wsdl:message>
<wsdl:message name="GetWorkbookConnectionsSoapOut">
  <wsdl:part name="parameters" element="tns:GetWorkbookConnectionsResponse"/>
</wsdl:message>
<wsdl:message name="GetWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:GetWorkbook"/>
</wsdl:message>
<wsdl:message name="GetWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:GetWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="NewWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:NewWorkbook"/>
</wsdl:message>
<wsdl:message name="NewWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:NewWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookExSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbookEx"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookExSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookExResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookForEditingSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbookForEditing"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookForEditingSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookForEditingResponse"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:OpenWorkbook"/>
</wsdl:message>
<wsdl:message name="OpenWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:OpenWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="RefreshDetachedSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshDetached"/>
</wsdl:message>
<wsdl:message name="RefreshDetachedSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshDetachedResponse"/>
</wsdl:message>
<wsdl:message name="RefreshExSoapIn">
  <wsdl:part name="parameters" element="tns:RefreshEx"/>
</wsdl:message>
<wsdl:message name="RefreshExSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshExResponse"/>
</wsdl:message>
<wsdl:message name="RefreshSoapIn">
  <wsdl:part name="parameters" element="tns:Refresh"/>
</wsdl:message>
<wsdl:message name="RefreshSoapOut">
  <wsdl:part name="parameters" element="tns:RefreshResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopyDetachedSoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyDetached"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopyDetachedSoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyDetachedResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopySoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopy"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookCopySoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookCopyResponse"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:SaveWorkbook"/>
</wsdl:message>
<wsdl:message name="SaveWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookResponse"/>
<wsdl:message name="SaveWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:SaveWorkbookResponse"/>
</wsdl:message>
<wsdl:message name="SetCalculationOptionsSoapIn">
  <wsdl:part name="parameters" element="tns:SetCalculationOptions"/>
</wsdl:message>
<wsdl:message name="SetCalculationOptionsSoapOut">
  <wsdl:part name="parameters" element="tns:SetCalculationOptionsResponse"/>
</wsdl:message>
<wsdl:message name="SetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetCellA1"/>
</wsdl:message>
<wsdl:message name="SetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetCellA1Response"/>
</wsdl:message>
<wsdl:message name="SetCellSoapIn">
  <wsdl:part name="parameters" element="tns:SetCell"/>
</wsdl:message>
<wsdl:message name="SetCellSoapOut">
  <wsdl:part name="parameters" element="tns:SetCellResponse"/>
</wsdl:message>
<wsdl:message name="SetParametersSoapIn">
  <wsdl:part name="parameters" element="tns:SetParameters"/>
</wsdl:message>
<wsdl:message name="SetParametersSoapOut">
  <wsdl:part name="parameters" element="tns:SetParametersResponse"/>
</wsdl:message>
<wsdl:message name="SetRangeA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetRangeA1"/>
</wsdl:message>
<wsdl:message name="SetRangeA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeA1Response"/>
</wsdl:message>
<wsdl:message name="SetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:SetRange"/>
</wsdl:message>
<wsdl:message name="SetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeResponse"/>
</wsdl:message>
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 updates to those products.

- Microsoft Office SharePoint Server 2007
- Microsoft SharePoint Server 2010
- Microsoft SharePoint Server 2013
- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 2.2.9.1: Office SharePoint Server 2007 does not support this error code.

<2> Section 2.2.9.2: Excel Services in Microsoft Office SharePoint Server 2010 and Excel Services in Microsoft Office SharePoint Server 2013 can use any of the values specified in section 2.2.9.1.

<3> Section 3.1.2.1: Excel Calculation Services 2007, Microsoft Excel Calculation Services 2010 and Microsoft Excel Calculation Services 2013 time out sessions (2) that are not edit sessions after a period of inactivity. This period of time is configurable.

<4> Section 3.1.2.2: Excel Services in Microsoft Office SharePoint Server 2007, Excel Services 2010 and Excel Services 2013 time out requests after a period of inactivity. This period of time is configurable.

<5> Section 3.1.2.2: In Excel Calculation Services 2007, Excel Calculation Services 2010 and Excel Calculation Services 2013, only some operations time out. Other operations can exceed the allotted time.

<6> Section 3.1.4.9: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

<7> Section 3.1.4.13: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

<8> Section 3.1.4.18: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

<9> Section 3.1.4.18: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the request.

<10> Section 3.1.4.18: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

<11> Section 3.1.4.18: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the request.
Section 3.1.4.19: Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [MSDN-UFES].

Section 3.1.4.19: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

Section 3.1.4.19: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

Section 3.1.4.19: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

Section 3.1.4.19: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

Section 3.1.4.21: Excel Services 2007, Excel Services 2010 and Excel Services 2013 have exceptions as noted in [MSDN-UFES].

Section 3.1.4.21: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

Section 3.1.4.21: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

Section 3.1.4.21: Excel Services 2007, Excel Services 2010 and Excel Services 2013 attempt to fall back to a supported language when an unsupported language is requested.

Section 3.1.4.21: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the language set for the site when none is specified in the current request.

Section 3.1.4.30: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

Section 3.1.4.32: Excel Services 2007, Excel Services 2010 and Excel Services 2013 set the values sent by the protocol client and do not return an error if rangeValues contains values in only one dimension (a single column or a single row), and if the first worksheet cell of rangeCoordinates falls on a workbook parameter.

Section 3.1.4.33: Excel Services 2007, Excel Services 2010 and Excel Services 2013 use the sheet (1) that was the active sheet when the workbook was opened.

Section 3.1.4.33: Excel Services 2007, Excel Services 2010 and Excel Services 2013 set the values sent by the protocol client and do not return an error if rangeValues contains values in only one dimension (a single column or a single row), and if the first worksheet cell of rangeCoordinates falls on a workbook parameter.
8 Change Tracking

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 42
Applicability 20
ArrayOfAnyType complex type 23
ArrayOfStatus complex type 23
ArrayOfString complex type 24
Attribute groups 28
Attributes 28

C
Capability negotiation 20
CellError simple type 26
Change tracking 178
Common data structures 29
faults 29
Common list of subcodes - fault 38
Complex types 23
ArrayOfAnyType 23
ArrayOfStatus 23
ArrayOfString 24
RangeCoordinates 24
Status 24
TypeExporter 25

D
Data model - abstract
server 42

E
Events
local - server 131
timer - server 131
Examples
get the value of a worksheet cell 132
overview 132
refresh data and retrieve a workbook example 135

F
Faults
common list of subcodes 38
Fields - vendor-extendable 21
Full WSDL 149

G
Get the value of a worksheet cell example 132
Glossary 13
Groups 28

I
Implementer - security considerations 148
Index of security parameters 148
Informative references 19
Initialization
server 42

L
List of subcodes – common data structures 29
Local events
server 131

M
Message processing
server 42
Messages
ArrayOfAnyType complex type 23
ArrayOfStatus complex type 23
ArrayOfString complex type 24
attribute groups 28
attributes 28
CellError simple type 26
common data structures 29
common list of subcodes - fault 38
complex types 23
elements 22
enumerated 22
faults 29
groups 28
namespaces 22
RangeCoordinates complex type 24
RangeReference simple type 26
SaveOptions simple type 27
SheetName simple type 27
simple types 25
Status complex type 24
syntax 22
transport 22
TypeExporter complex type 25
WorkbookType simple type 28

N
Namespaces 22
Normative references 17

O
Operations
Calculate 44
CalculateA1 46
CancelWorkbook 49
CancelRequest 52
CloseWorkbook 54
EnsureWorkbookModel 56
GetApiVersion 59
GetCell 61
GetCellA1 63
GetChartImageUrl 66
GetPublishedItemNames 69
GetRange 72
GetRangeA1 75
GetSessionInformation 78
GetSheetNames 81
GetWorkbook 85

[MS-ESP] - v20190618
Excel Web Services Protocol
Copyright © 2019 Microsoft Corporation
Release: June 18, 2019