[MS-ESP]:
Excel Web Services Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

Revision Summary

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

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

Table of Contents

[MS-ESP] — v20120410
Excel Web Services Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: Tuesday, April 10, 2012

[MS-ESP] — v20120410
Excel Web Services Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: Tuesday, April 10, 2012

[MS-ESP] — v20120410
Excel Web Services Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: Tuesday, April 10, 2012

10 / 150

1 Introduction

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

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

1.1 Glossary

The following terms are defined in [MS-GLOS]:

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

The following terms are defined in [MS-OFCGLOS]:

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

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

The following terms are specific to this document:

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

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

1.2 References

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

1.2.1 Normative References

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

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

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

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

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

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

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

[ODF] OASIS, "Open Document Format for Office Applications (OpenDocument) v1.1", 1 February 2007, http://www.oasis-open.org/specs/#opendocumentv1.1

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

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

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

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

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

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

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

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

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

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

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

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

1.2.2   Informative References

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

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

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

[MSFT-ACRR] Microsoft Corporation, "About Cell and Range References", http://office.microsoft.com/en-us/help/HP051983231033.aspx

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

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

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

1.3   Protocol Overview (Synopsis)

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

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

- Open existing workbooks for viewing and editing and close open workbooks.

- Create new workbooks.

- Save workbooks.

- Retrieve the full contents of the current open copy.

- Recalculation.

- **Refresh connected data**.

- Set specific values into a **worksheet cell** or **range**.

- Get specific values from a worksheet cell or range.

- Cancel a request.

- Get the names of all the **published items** in a workbook.

- Get a **URL** to an image of a chart in a workbook.

- Get information about a session.

- Get the version of the protocol server.

1.3.1   Published Items and Results Returned by this Protocol

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

1.4   Relationship to Other Protocols

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

The following diagram shows the underlying messaging and transport stack used by the protocol:
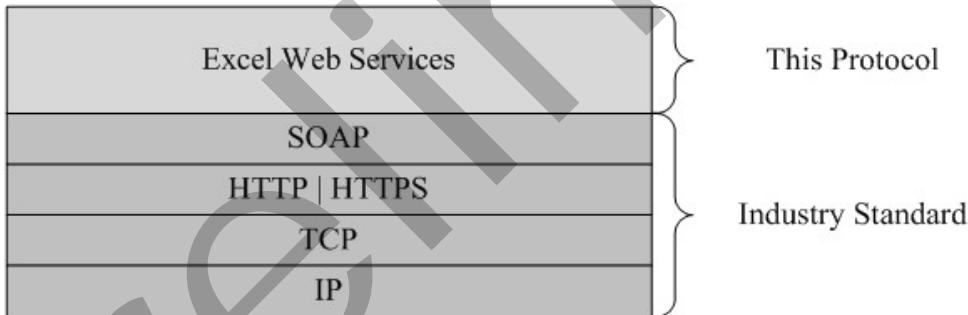
Figure 1: This protocol in relation to other protocols

1.5   Prerequisites/Preconditions

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

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

1.6   Applicability Statement

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

1.7   Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

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

- **Protocol versions:** This protocol is not versioned.

- **Capability negotiation:** This protocol does not support capability negotiation.

1.8   Vendor-Extensible Fields

None.

1.9   Standards Assignments

None.

2 Messages

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

2.1 Transport

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

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

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

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type	Description
ArrayOfAnyType	An array of **anyType** elements.
ArrayOfStatus	An array of status information.
RangeCoordinates	Specifies a range on a workbook.
Status	The status information returned by the protocol server.
TypeExporter	MUST NOT be used.

2.2.4.1 ArrayOfAnyType

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

```
<xs:complexType name="ArrayOfAnyType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="anyType" nillable="true"/>
  </xs:sequence>
</xs:complexType>
```

anyType: An **anyType** value.

2.2.4.2 ArrayOfStatus

An array of status information.

- The protocol server MUST use the **Status** complex type to return informative status messages to the protocol client for operations that succeed.

- The protocol client MUST NOT perform any logic based on the specific values in the **Name** or **Message** elements returned in the **Status** type.

This complex type is defined as follows.

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

Status: The status information returned by the protocol server.

2.2.4.3 RangeCoordinates

Specifies a range on a workbook.

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

- ▪ If **Height** is 0, then **Row**, **Column**, and **Width** MUST be 0.

- ▪ If **Width** is 0, then **Row**, **Column**, and **Height** MUST be 0.

This complex type is defined as follows.

```
<xs:complexType name="RangeCoordinates">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Row" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Column" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

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

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

Height: The number of rows (2). MUST be greater than or equal to 0. The sum of **Row** and **Height** MUST NOT be greater than 1,048,576.

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

2.2.4.4 Status

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

```
<xs:complexType name="Status">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Severity" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

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

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

Value	Meaning
Error	The status is of severity error.

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

Message: The message description.

2.2.4.5 TypeExporter

MUST NOT be used. It is defined as follows.

```
<xs:complexType name="TypeExporter">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="CellError" type="tns:CellError"/>
  </xs:sequence>
</xs:complexType>
```

CellError: MUST NOT be used.

2.2.5 Simple Types

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

Simple type	Description
CellError	The **CellError** type is an enumeration of a set of **cell error values** that a worksheet cell can contain.
RangeReference	A string that is used by the protocol server to identify a range on a **sheet** in a workbook, in some of the operations specified by this protocol.
SheetName	A string that is used by the protocol server to find a sheet in a workbook in some of the operations specified in this protocol.
WorkbookType	An enumeration that specifies the form of a requested workbook.

2.2.5.1 CellError

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

```
<xs:simpleType name="CellError">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Div0"/>
    <xs:enumeration value="NA"/>
    <xs:enumeration value="Name"/>
    <xs:enumeration value="Null"/>
    <xs:enumeration value="Num"/>
    <xs:enumeration value="Ref"/>
    <xs:enumeration value="Value"/>
  </xs:restriction>
```

```
</xs:simpleType>
```

The following table specifies the allowable values for CellError:

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

2.2.5.2   RangeReference

A string that is used by the protocol server to identify a range on a sheet in a workbook, in some of the operations specified by this protocol. It has the following specification:

▪ MUST NOT be longer than 32768 characters.

▪ SHOULD use the A1 reference style as defined in [MSFT-ACRR], but MAY use any format supported by the server to identify the range.

This simple type is defined as follows.

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

2.2.5.3   SheetName

A string that is used by the protocol server to find a sheet in a workbook in some of the operations specified in this protocol. It has the following specification:

▪ MUST NOT be longer than 31 characters.

▪ If empty, the protocol server MUST use the first sheet in the workbook unless specified otherwise in the operation.

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

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

This simple type is defined as follows.

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

2.2.5.4 WorkbookType

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

This enumeration is defined as follows.

```
<xs:simpleType name="WorkbookType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="FullWorkbook"/>
    <xs:enumeration value="FullSnapshot"/>
    <xs:enumeration value="PublishedItemsSnapshot"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for WorkbookType:

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

2.2.9 Faults

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

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

2.2.9.1 List of Subcodes

The following table describes the error codes.

Value	Meaning
Blank	MUST NOT be used.
InternalError	An unexpected error occurred in the protocol server.
NotMemberOfRole	The user does not have **permissions** to complete the operation.
EncryptionTooLow	The protocol server is configured to accept only HTTPS, and the protocol client has tried to connect with a non HTTPS.
ConfirmDataRefresh	A connected data refresh **confirmation** is requested by the protocol client before connected data refresh on a workbook.
ExternalDataRefreshFailed	Connected data refresh failed.
ExternalDataRefreshFailedReal	Connected data refresh failed.
ExternalDataRefreshFailedReadOrViewOnly	Connected data refresh failed.
NoAccessToOdcReadOrViewOnly	Connected data refresh failed.
ErrorConnectingToDataSourceReadOrViewOnly	Connected data refresh failed.
InvalidUserNameOrPasswordReadOrViewOnly	Connected data refresh failed.
InvalidUserNameOrPassword	Connected data refresh failed.
ExternalDatatNotAllowedOnTL	Connected data refresh failed.
InvalidUriorLocalConnectionPath	Connected data refresh failed.
DclNotTrusted	Connected data refresh failed.
OdcFileDoesNotExist	Connected data refresh failed.
ErrorAccessingSsofromWfe	Connected data refresh failed.
ErrorAccessingSsoFromEcs	Connected data refresh failed.
SsoError	Connected data refresh failed.
ErrorConnectingToDataSource	Connected data refresh failed.
UnattendedAccountNotConfigured	Connected data refresh failed.
IncorrectAuthUsed	Connected data refresh failed.
ExternalDataRefreshFailed_Details	Connected data refresh failed.
PivotNoOverlapAllowed	A **PivotTable** grew to the point where it would overlap another PivotTable.
PivotOverlapList	A PivotTable grew to the point where it would overlap a

Value	Meaning
	list (2).
PivotCubeNotUnique	An **OLAP query** found more than one **cube** of the same name.
NoDataItemsOlap	An OLAP query found nothing to display.
OlapSchemaFail	An OLAP query was unable to fetch the schema of the cube.
InvalidPivot	A PivotTable is invalid.
OlapNoMoreDetail	A PivotTable has reached the lowest **level** of detail.
DrillActionInvalid	A **PivotTable member** cannot be expanded any further.
DriverNotOlap	The database does not support OLAP.
CubeMissing	The OLAP query was unable to locate the cube in the database.
SkipLine	Not all data received from the OLAP query could be displayed.
CDimMismatch	The OLAP **data provider** returned an unexpected number of **dimensions (2)**.
DataSetAxisTooBig	The row (2) or column (2) area of the PivotTable is too large.
CubeIntegrityError	There was a problem with the structure of the OLAP cube.
PivotNotExpandedtoProtectCellContents	The PivotTable would have filled worksheet cells that already contain values, so the operation was canceled.
PivotTruncatedSheetEdge	The PivotTable was truncated by a worksheet edge.
DdlGrpNotSup	The PivotTable operation is not supported by the OLAP data provider.
PivotFormulasCollided	Some calculated PivotTable members were removed from the PivotTable because new PivotTable members of the same name appeared during refresh.
NoRetItems	The query of the **PivotTable page field** did not return any items.
OlapConnectVisualModeProvider	The PivotTable operation is not supported by the OLAP data provider.
PivotNoSourceData	The PivotTable was saved without the underlying data.
PivotFilteringError	The PivotTable could not be filtered.
PivotTableIncorrectDateFormat	An invalid date was used to **filter** the PivotTable.
PivotTableEndingLargerThanStarting	An invalid pair of starting and ending values was used to filter the PivotTable.

Value	Meaning
PivotTableIncorrectNumberFormat	A non-numeric value was used to filter the PivotTable, when a numeric value was required.
PivotTableOutOfRange	An out of range value was used to filter the PivotTable.
MergedCellOverlap	The PivotTable could not **grow** over part of a **merged cell**.
PivotSourceNotAvailable	The PivotTable could not be refreshed because its source is missing.
UnexpectedPivotError	The PivotTable has encountered an unexpected error.
EditModeCannotAcquireLockUnknownReason	An attempt to acquire or refresh a lock on a file failed.
EditModeCannotAcquireLock	An attempt to acquire or refresh a lock on a file failed because the file is already **locked**.
EditModeCannotAcquireLockUnknownUser	An attempt to acquire or refresh a lock on a file failed because the file is locked by an unknown user.
EditModeWorkbookCheckedOutRequired	An attempt to acquire or refresh a lock on a file failed because the file is not **checked out**.
EditModeWorkbookCheckedOutToLocalDrafts	An attempt to acquire or refresh a lock on a file failed because the file is checked out to a local drafts folder.
SiteTooBusy	MUST NOT be used.
SiteCollectionMemoryQuotaExceeded	MUST NOT be used.
SessionMemoryQuotaExceeded	MUST NOT be used.
SiteCollectionChartQuotaExceeded	MUST NOT be used.
RequestLimitsExceeded	Too many cells requested.
SavingUnsupportedInViewMode	An attempt was made to save a workbook from a **view session**.
SavingWorkbookNoWorkbookName	An attempt was made to save a workbook from a **new workbook session**.
FileSaveGeneric	The save operation failed.
SavingWorkbookInvalidPermissions	The user does not have permissions to save to the specified location.
SavingWorkbookMaximumSizeExceeded	An attempt to save a workbook failed because the workbook exceeds the maximum permitted size.
SavingWorkbookLockedOrCheckedOut	An attempt to save a workbook failed because the file is locked or checked out.
SavingWorkbookNotCheckedOut	An attempt to save a workbook failed because the file is not checked out.
SavingWorkbookAccessDenied	An attempt to save a workbook failed because the user does not have permission to overwrite an existing file.

Value	Meaning
SavingWorkbookInvalidFileType	MUST NOT be used.
SavingWorkbookFileNameExists	An attempt to save a workbook failed because the file already exists and the option to overwrite existing files was not specified.
SavingWorkbookNotTrustedLocation	An attempt to save a workbook failed because of the security settings on the protocol server.
SavingWorkbookInvalidLocation	An attempt to save a workbook failed because the specified file location is invalid.
SavingWorkbookInvalidFileName	An attempt to save a workbook failed because the specified file name is invalid.
SavingWorkbookTrustedLocationSettingWarning	An attempt to save a workbook failed because of the security settings on the protocol server.
SavingWorkbookAlmostTooLarge	MUST NOT be used.
SavingWorkbookTooLarge	MUST NOT be used.
NewWorkbookInvalidLocation	An attempt to create a new workbook failed because the specified location is invalid.
FileOpenSecuritySettings	Cannot open workbook because of security settings.
FileOpenNotFound	The specified workbook file does not exist.
FileOpenAccessDenied	The user does not have access to the workbook file.
FileCorrupt	The workbook file is in a format not recognized by the protocol server.
OpenWorkbookTooLarge	The protocol server failed to create a session (2) for a workbook because it is too large.
OpenWorkbookLocationNotTrusted	The protocol server failed to create a session (2) for a workbook because of the security settings on the protocol server.
OpenWorkbookAccessDenied	The protocol server failed to create a session (2) for a workbook because the user does not have sufficient permissions.
WorkbookNotSupported	The protocol server failed to create a session (2) for this workbook because it has unsupported content.
WorkbookNotSupportedViewRight	The protocol server failed to create a session (2) for this workbook because it has unsupported content and the user does not have Open Item permission.
UnsupportedFeatureViewWarning	MUST NOT be used.
UnsupportedFeatureSaveCopyWarning	MUST NOT be used.
ExternalWorkbookLinksDisabled	MUST NOT be used.
WorkbookLockedRead	An attempt to open a workbook failed because a file

Value	Meaning
	lock prevents it from being read.
EditModeCannotEditNonSharePointWorkbook	An attempt to open an **edit session** failed because the file is not in a **document library**.
EditModeAccessDenied	An attempt to open a workbook in an edit session failed because the user does not have sufficient permissions to edit the **workbook file**.
GenericFileOpenError	An unspecified error occurred when opening the workbook file.
IRMedWorkbook	An attempt to open a workbook failed because the workbook is in an **Information Rights Management (IRM)** document library.
ParameterInvalid	An attempt to open a workbook failed because the workbook contains one or more invalid **workbook parameters**.
ParameterLimitExceeded	An attempt to open a workbook failed because the workbook contains too many workbook parameters.
WorkbookInternalLimit	The workbook cannot be loaded by the protocol server because of a file-size limitation.
OldFormatConversionFailed	MUST NOT be used.
EditModeWorkbookExternalData	An attempt to open a workbook as an edit session failed because it contains connected data or **user-defined functions (UDFs)**, and the workbook file has not been checked out.
FormulaEditingNotEnabled	The operation failed because it tried to set a formula into a workbook parameter.
DimensionAndArrayMismatch	An attempt to set a range failed because the dimensions (2) of the specified array of cells do not match the specified range.
SetRangeFailure	The operation failed to set a value into a cell.
ObjectTypeNotSupported	The operation failed because it tried to update a cell with an object of an unsupported type.
ParameterDoesNotExist	The specified workbook parameter does not exist.
FormulaParseFailed	The specified formula could not be parsed.
SetParameterFailure	An unspecified error occurred when attempting to set a workbook parameter.
CannotEditPivotTable	The operation failed because it tried to set a cell in a PivotTable.
InsDelCellsNonblankCells	The operation failed because it would require shifting non-blank cells off the sheet.
InsDelCellsArrayFmla	The operation failed because it tried to change part of

Value	Meaning
	an **array formula**.
InsDelCellsPivotTable	The operation failed because it tried to change part of a PivotTable.
InsDelCellsTable	The operation failed because it tried to change part of a **table**.
InsDelCellsMergedCells	The operation failed because it tried to change part of a merged cell.
InsDelCellsDatatable	The operation failed because it tried to change part of a **data table**.
InsDelCellsRowsForWholeColumn	The operation failed because it tried to insert or delete a whole row (2), but the specified range is a whole column (2).
InsDelCellsColumnsForWholeRow	The operation failed because it tried to insert or delete a whole column (2), but the specified range is a whole row (2).
InvalidSheetName	The operation failed because the specified sheet name does not exist.
RangeParseError	The operation failed because the specified range is invalid.
SpecifiedRangeNotFound	The operation failed because the specified range was not found.
RangeDefinitionError	The operation failed because the specified range is not a valid sheet location.
DiscontiguousRangeNotSupported	The operation failed because a **non-contiguous range** was specified.
RangeRequestAreaExceeded	The requested range was too large.
SheetRangeMismatch	The sheet provided as the sheet argument is not the same as the sheet specified in the range argument.
ImageGenFailed	The protocol server failed to create an image.
ApiInvalidArgument	The operation failed because an argument that passed was invalid.
ApiInvalidCoordinate	MUST NOT be used.
ConfirmationUnsupported	MUST NOT be used.
ApiInvalidDimensionReturned	MUST NOT be used.
ApiMissingSiteContext	MUST NOT be used.
Ewr_IllegalPageSize	MUST NOT be used.
Ewr_WorkbookNotSpecified	MUST NOT be used.
Ewr_LoadProgress	MUST NOT be used.

Value	Meaning
Ewr_FileCorrupt	MUST NOT be used.
Ewr_WorkbookNotSupported	MUST NOT be used.
Ewr_GenericFileOpenError	MUST NOT be used.
Ewr_InvalidParameterGeneric	MUST NOT be used.
Ewr_InvalidParameterMandatoryMissing	MUST NOT be used.
Ewr_InvalidParameterUnexpectedFormat	MUST NOT be used.
Ewr_InvalidParameterOutOfRange	MUST NOT be used.
Ewr_InvalidParameterOperationNotAllowed	MUST NOT be used.
Ewr_ParameterInvalid	MUST NOT be used.
Ewr_ParameterLimitExceeded	MUST NOT be used.
Ewr_ParameterDoesNotExist	MUST NOT be used.
Ewr_SetParameterFailure	MUST NOT be used.
Ewr_SetParameterFailureOnOpen	MUST NOT be used.
Ewr_ParameterReadFromDocPropsError	MUST NOT be used.
Ewr_FormulaEditingNotEnabled	MUST NOT be used.
Ewr_InvalidVisibleItem	MUST NOT be used.
Ewr_EditSaveAsOverCurrentWorkbook	MUST NOT be used.
Ewr_RequestTimedOut	MUST NOT be used.
Ewr_ServerCommunicationError	MUST NOT be used.
Ewr_ServerAuthenticationError	MUST NOT be used.
Mewa_PageStateInitFailed	MUST NOT be used.
Mewa_CriticalParameterMissing	MUST NOT be used.
Mewa_GridViewRenderingFailed	MUST NOT be used.
Mewa_FeatureNotActive	MUST NOT be used.
OperationNotSupportedInViewOnly	The operation failed because it is not supported in a view session.
OperationOnInvalidState	An attempt was made to open a session (2) that already has a workbook assigned.
InvalidOrTimedOutSession	The operation failed because the session (2) is invalid or timed out.
ScratchPadChangesLost	A view session (2) that is not in **view-only mode** timed out while the user was attempting to convert it to

Value	Meaning
	an edit session.
MaxSessionsPerUserExceeded	The user has exceeded the maximum allowed session (2) count by the protocol server.
OperationCanceled	The user canceled a request that was being processed by a session (2).
MaxRequestDurationExceeded	The request failed because it took too long to process.
NoOccurrencesFound	The protocol server could not match the sought string to any worksheet cell relevant to the search.
EcsInvalidNamedObject	An invalid **named object** was requested.
NoPublishedItems	The workbook does not have any **published items**.
RetryError	The operation could not be finished at this time, but the request may succeed if it is sent again by the protocol client.
StopOpenOnRefreshFailure	An attempt to open a workbook failed because connected data refresh failed.
ProductExpired	The protocol server returns this error when the application **license key** is invalid.
SortActionInvalid	Sorting an **AutoFilter** failed because it contains merged cells that are not all the same size.
SortActionNotSupportedWithPivot	Sorting an AutoFilter failed because it contains a PivotTable.
CircularReferencesFound	MUST NOT be used.
SessionNotFound	MUST NOT be used.
NewWorkbookSessionTimeout	A new workbook session has timed out.
EditXLSFile	MUST NOT be used.
UnsupportedFeatureEditWarning	Edit session is not supported for this workbook.
EditModeSessionExists	The view session could not be saved because the workbook is open in an edit session.
NewerWorkbookExists	The view session could not be saved because the workbook has been modified since the session (2) was opened.
ListEntryNotFound_1	MUST NOT be used.
ListEntryNotFound_2	MUST NOT be used.
ListEntryAlreadyExists_1	MUST NOT be used.
ListEntryAlreadyExists_2	MUST NOT be used.
InvalidArgumentValue	MUST NOT be used.

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

2.2.9.2 Common List of Subcodes

In the event of an application error during an operation the protocol server returns a SOAP fault response, which SHOULD<1> contain one of the following subcodes:

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

3 Protocol Details

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

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

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

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

The following high-level sequence diagram illustrates the operation of the protocol.
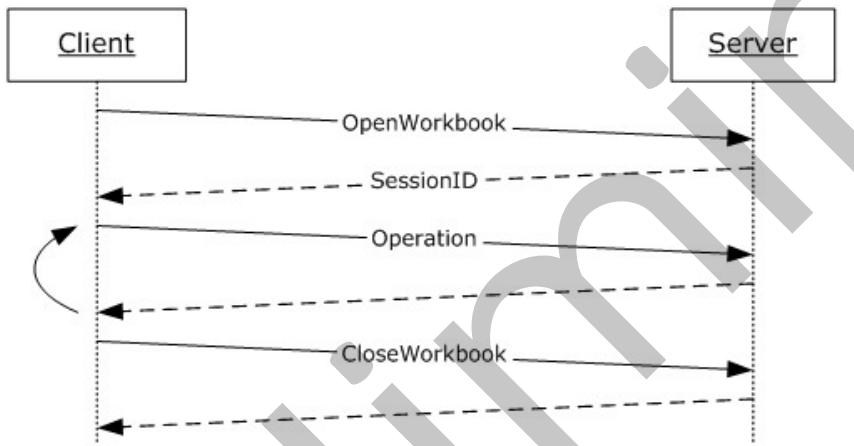
Figure 2: High-level sequence diagram

Session Behavior

View Sessions

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

New Workbook Sessions

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

Edit Sessions

An edit session is created on a successful **OpenWorkbookForEditing** request, unless one already exists for the same **workbookPath**, as specified in OpenWorkbookForEditing (section 3.1.4.18.2.1). A new participant is added to the edit session on a successful **OpenWorkbookForEditing** request.

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

Request Behavior

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

3.1 Server Details

3.1.1 Abstract Data Model

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

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

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

3.1.2 Timers

3.1.2.1 Session Timeout

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

3.1.2.2 Request Timeout

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

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

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

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

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

WSDL operation	Description
Calculate	This operation is used to calculate a specific range in the workbook.
CalculateA1	This operation is used to calculate a specific range in the workbook.
CalculateWorkbook	This operation is used to calculate the workbook.
CancelRequest	This operation is used to cancel an operation that is currently running on the protocol server.
CloseWorkbook	This operation is used to close an existing session (2) or leave an edit session on the protocol server.
GetApiVersion	This operation is used to get the version of the protocol server.
GetCell	This operation is used to retrieve the contents of a specific cell in the workbook.
GetCellA1	This operation is used to retrieve the contents of a specific cell in the workbook.
GetRange	This operation is used to retrieve the contents of a **named range** in the workbook.
GetRangeA1	This operation is used to retrieve the contents of a named range in the workbook.
GetSessionInformation	This operation retrieves information about a session (2) on the protocol server.
GetWorkbook	This operation is used to retrieve a workbook file.
OpenWorkbook	This operation is used to create a new view session, giving access to a workbook on the protocol server.
Refresh	This operation is used to refresh connected data on a workbook.
SetCell	This operation is used to set a value to a specific worksheet cell in the workbook.
SetCellA1	This operation is used to set a value to a specific worksheet cell in the workbook.
SetRange	This operation is used to set values to a range of worksheet cells on a workbook.
SetRangeA1	This operation is used to set values to a range of worksheet cells on a workbook.

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

WSDL operation	Description
GetChartImageUrl	This operation returns the URL of a chart in a workbook.
GetPublishedItemNames	This operation returns a list of the published items in a workbook.
GetSheetNames	The operation returns information about the sheets in a workbook.
NewWorkbook	This operation is used to create a new workbook session.
OpenWorkbookForEditing	This operation is used to create a new edit session, giving access to a workbook on the protocol server.
SaveWorkbook	This operation is used to save a workbook associated with an open session (2) to the same location from which it was opened.
SaveWorkbookCopy	This operation is used to save a workbook associated with an open session (2) to a specified location, and in a specified format.
SetCalculationOptions	This operation is used to set the workbook **calculation mode**.
SetParameters	This operation to sets values of workbook parameters in a workbook.

3.1.4.1 Calculate

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

```
<wsdl:operation name="Calculate">
  <wsdl:input message="tns:CalculateSoapIn"/>
  <wsdl:output message="tns:CalculateSoapOut"/>
</wsdl:operation>
```

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

1. If the protocol determines that the workbook doesn't need to be calculated it MUST succeed the operation and it MUST ignore the **sheetName** and **rangeCoordinates** parameters.

2. **sheetName** MUST NOT be empty if **rangeCoordinates** is not empty, as defined in RangeCoordinates. If **sheetName** is empty, the whole workbook MUST be calculated.

3. If **rangeCoordinates** is empty, as defined in section 2.2.4.3, and the **sheetName** is not empty, the whole sheet, which was specified in **sheetName**, MUST be calculated.

4. If **rangeCoordinates** is not empty, the specified range MUST be calculated.

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject

Subcode
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.1.1 Messages

3.1.4.1.1.1 CalculateSoapIn

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

The **SOAP action** value is:

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

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

3.1.4.1.1.2 CalculateSoapOut

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

The SOAP action value is:

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

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

3.1.4.1.2 Elements

3.1.4.1.2.1 Calculate

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

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

sessionId: A **session identifier (2)**. MUST be present in the request message. MUST NOT be longer than 1023 characters. SHOULD have a value that has been obtained from a previous call to

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

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

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

3.1.4.1.2.2 CalculateResponse

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

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

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

3.1.4.1.3 Complex Types

None.

3.1.4.1.4 Simple Types

None.

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.4.2 CalculateA1

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

```
<wsdl:operation name="CalculateA1">
  <wsdl:input message="tns:CalculateA1SoapIn"/>
  <wsdl:output message="tns:CalculateA1SoapOut"/>
</wsdl:operation>
```

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

1. If the protocol determines that the workbook doesn't need to be calculated it MUST succeed the operation and it MUST ignore the **sheetName** and **rangeName** parameters.

2. **sheetName** MUST be present and MUST NOT be empty, if **rangeName** is not empty. If **sheetName** is not present or is empty, the whole workbook MUST be calculated.

3. **rangeName** MUST be present. If **rangeName** is empty, and **sheetName** is present and not empty, the whole sheet, which was specified in **sheetName**, MUST be calculated.

4. If **rangeName** is not empty, the specified range MUST be calculated.

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.2.1 Messages

3.1.4.2.1.1 CalculateA1SoapIn

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

The SOAP action value is:

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

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

3.1.4.2.1.2 CalculateA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.2.2 Elements

3.1.4.2.2.1 CalculateA1

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

```
<xs:element name="CalculateA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

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

3.1.4.2.2.2 CalculateA1Response

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

```
<xs:element name="CalculateA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

3.1.4.2.3 Complex Types

None.

3.1.4.2.4 Simple Types

None.

3.1.4.2.5 Attributes

None.

3.1.4.2.6 Groups

None.

3.1.4.2.7 Attribute Groups

None.

3.1.4.3 CalculateWorkbook

This operation is used to calculate the workbook defined as follows.

```
<wsdl:operation name="CalculateWorkbook">
  <wsdl:input message="tns:CalculateWorkbookSoapIn"/>
  <wsdl:output message="tns:CalculateWorkbookSoapOut"/>
</wsdl:operation>
```

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

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.3.1 Messages

3.1.4.3.1.1 CalculateWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.3.1.2 CalculateWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.3.2 Elements

3.1.4.3.2.1 CalculateWorkbook

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

```
<xs:element name="CalculateWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="calculateType" type="tns:CalculateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

3.1.4.3.2.2 CalculateWorkbookResponse

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

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

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

3.1.4.3.3 Complex Types

None.

3.1.4.3.4 Simple Types

3.1.4.3.4.1 CalculateType

The type of calculation to be executed.

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

The following table specifies the allowable values for CalculateType:

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

3.1.4.3.5 Attributes

None.

3.1.4.3.6 Groups

None.

3.1.4.3.7 Attribute Groups

None.

3.1.4.4 CancelRequest

This operation is used to cancel an operation that is currently running on the protocol server and is defined as follows.

```
<wsdl:operation name="CancelRequest">
  <wsdl:input message="tns:CancelRequestSoapIn"/>
  <wsdl:output message="tns:CancelRequestSoapOut"/>
</wsdl:operation>
```

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

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

- The following MUST NOT cause an application error in the **CancelRequest** operation:

  ▪ An operation is not currently running on the session (2).

44 / 150

▪A currently running operation on the session (2) cannot be cancelled.

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

3.1.4.4.1 Messages

3.1.4.4.1.1 CancelRequestSoapIn

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

The SOAP action value is:

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

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

3.1.4.4.1.2 CancelRequestSoapOut

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

The SOAP action value is:

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

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

3.1.4.4.2 Elements

3.1.4.4.2.1 CancelRequest

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

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

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

3.1.4.4.2.2 CancelRequestResponse

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

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

45 / 150

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

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

3.1.4.4.3 Complex Types

None.

3.1.4.4.4 Simple Types

None.

3.1.4.4.5 Attributes

None.

3.1.4.4.6 Groups

None.

3.1.4.4.7 Attribute Groups

None.

3.1.4.5 CloseWorkbook

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

```
<wsdl:operation name="CloseWorkbook">
  <wsdl:input message="tns:CloseWorkbookSoapIn"/>
  <wsdl:output message="tns:CloseWorkbookSoapOut"/>
</wsdl:operation>
```

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

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

3.1.4.5.1 Messages

3.1.4.5.1.1 CloseWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.5.1.2 CloseWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.5.2 Elements

3.1.4.5.2.1 CloseWorkbook

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

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

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

3.1.4.5.2.2 CloseWorkbookResponse

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

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

status: MUST NOT be present.

3.1.4.5.3 Complex Types

None.

3.1.4.5.4 Simple Types

None.

3.1.4.5.5 Attributes

None.

3.1.4.5.6 Groups

None.

3.1.4.5.7 Attribute Groups

None.

3.1.4.6 GetApiVersion

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

```
<wsdl:operation name="GetApiVersion">
  <wsdl:input message="tns:GetApiVersionSoapIn"/>
  <wsdl:output message="tns:GetApiVersionSoapOut"/>
</wsdl:operation>
```

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

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

3.1.4.6.1 Messages

3.1.4.6.1.1 GetApiVersionSoapIn

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

The SOAP action value is:

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

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

3.1.4.6.1.2 GetApiVersionSoapOut

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

The SOAP action value is:

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

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

3.1.4.6.2 Elements

3.1.4.6.2.1 GetApiVersion

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

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

3.1.4.6.2.2 GetApiVersionResponse

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

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

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

status: MUST NOT be present.

3.1.4.6.3 Complex Types

None.

3.1.4.6.4 Simple Types

None.

3.1.4.6.5 Attributes

None.

3.1.4.6.6 Groups

None.

3.1.4.6.7 Attribute Groups

None.

3.1.4.7 GetCell

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

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

```
  <wsdl:input message="tns:GetCellSoapIn"/>
  <wsdl:output message="tns:GetCellSoapOut"/>
</wsdl:operation>
```

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

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

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

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

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.7.1 Messages

3.1.4.7.1.1 GetCellSoapIn

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

The SOAP action value is:

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

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

3.1.4.7.1.2 GetCellSoapOut

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

The SOAP action value is:

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

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

3.1.4.7.2   Elements

3.1.4.7.2.1   GetCell

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

```
<xs:element name="GetCell">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

sheetName: The name of the sheet that contains the cell. See section 2.2.5.3. MUST be present. MUST NOT be empty.

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

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

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

3.1.4.7.2.2   GetCellResponse

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

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

GetCellResult: The returned cell value.

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

3.1.4.7.3 Complex Types

None.

3.1.4.7.4 Simple Types

None.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groupos

None.

3.1.4.8 GetCellA1

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

```
<wsdl:operation name="GetCellA1">
  <wsdl:input message="tns:GetCellA1SoapIn"/>
  <wsdl:output message="tns:GetCellA1SoapOut"/>
</wsdl:operation>
```

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

- The **rangeName** specifies the location of the sheet cell to retrieve. The sheet that contains the sheet cell to retrieve is determined in the following manner.

- If **rangeName** specifies a sheet, that is the sheet which MUST be used.

- Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section 2.2.5.3.

- If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY<6> use any sheet in the workbook.

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

- If **formatted** is true, the sheet cell returned as the **GetCellA1Result** element MUST be type **string**, and MUST be number formatted according to how the sheet cell is number formatted in the workbook.

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.8.1 Messages

3.1.4.8.1.1 GetCellA1SoapIn

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

The SOAP action value is:

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

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

3.1.4.8.1.2 GetCellA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.8.2 Elements

3.1.4.8.2.1 GetCellA1

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

```
<xs:element name="GetCellA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
```

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

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

sheetName: The name of the sheet that contains the cell. See section 2.2.5.3 for specification. MUST be present.

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

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

3.1.4.8.2.2 GetCellA1Response

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

```
<xs:element name="GetCellA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetCellA1Result"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetCellA1Result: The returned cell value.

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

3.1.4.8.3 Complex Types

None.

3.1.4.8.4 Simple Types

None.

3.1.4.8.5 Attributes

None.

3.1.4.8.6 Groups

None.

3.1.4.8.7 Attribute Groups

None.

3.1.4.9 GetChartImageUrl

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

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

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

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

3.1.4.9.1 Messages

3.1.4.9.1.1 GetChartImageUrlSoapIn

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

The SOAP action value is:

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

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

3.1.4.9.1.2 GetChartImageUrlSoapOut

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

The SOAP action value is:

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

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

3.1.4.9.2 Elements

3.1.4.9.2.1 GetChartImageUrl

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

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

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

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

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

3.1.4.9.2.2 GetChartImageUrlResponse

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

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

GetChartImageUrlResult: The URL of the requested chart.

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

3.1.4.9.3 Complex Types

3.1.4.9.3.1 Size

Specifies the size of a chart image defined as follows.

```
<xs:complexType name="Size">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int"/>
  </xs:sequence>
</xs:complexType>
```

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

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

3.1.4.9.4 Simple Types

None.

3.1.4.9.5 Attributes

None.

3.1.4.9.6 Groups

None.

3.1.4.9.7 Attribute Groups

None.

3.1.4.10 GetPublishedItemNames

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

```
<wsdl:operation name="GetPublishedItemNames">
  <wsdl:input message="tns:GetPublishedItemNamesSoapIn"/>
  <wsdl:output message="tns:GetPublishedItemNamesSoapOut"/>
</wsdl:operation>
```

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

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

3.1.4.10.1 Messages

3.1.4.10.1.1 GetPublishedItemNamesSoapIn

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

The SOAP action value is:

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

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

3.1.4.10.1.2 GetPublishedItemNamesSoapOut

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

The SOAP action value is:

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

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

3.1.4.10.2 Elements

3.1.4.10.2.1 GetPublishedItemNames

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

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

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

3.1.4.10.2.2 GetPublishedItemNamesResponse

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

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

GetPublishedItemNamesResult: An array of **WorkbookItem** elements as specified in section 3.1.4.10.3.1. MUST be present.

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

3.1.4.10.3 Complex Types

3.1.4.10.3.1 ArrayOfWorkbookItem

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

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

WorkbookItem: A **WorkbookItem** as specified in section 3.1.4.10.3.2.

3.1.4.10.3.2 WorkbookItem

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

```
<xs:complexType name="WorkbookItem">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:ItemType"/>
  </xs:sequence>
</xs:complexType>
```

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

Type: The type of the published item as specified in section 3.1.4.10.4.1.

3.1.4.10.4 Simple Types

3.1.4.10.4.1 ItemType

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

```
<xs:simpleType name="ItemType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Chart"/>
    <xs:enumeration value="PivotTable"/>
    <xs:enumeration value="Table"/>
    <xs:enumeration value="NamedRange"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for ItemType:

Value	Meaning
Chart	Specifies that the published item is a chart.
PivotTable	Specifies that the published item is a PivotTable.
Table	Specifies that the published item is a list (2).
NamedRange	Specifies that the published item is a named range.

3.1.4.10.5 Attributes

None.

3.1.4.10.6 Groups

None.

3.1.4.10.7 Attribute Groups

None.

3.1.4.11 GetRange

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

```
<wsdl:operation name="GetRange">
  <wsdl:input message="tns:GetRangeSoapIn"/>
  <wsdl:output message="tns:GetRangeSoapOut"/>
</wsdl:operation>
```

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

1. The **rangeCoordinates** on the sheet specified by **sheetName**, specify the location of the sheet cells to retrieve.

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

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

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

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

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.11.1 Messages

3.1.4.11.1.1 GetRangeSoapIn

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

The SOAP action value is:

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

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

3.1.4.11.1.2 GetRangeSoapOut

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

The SOAP action value is:

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

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

3.1.4.11.2 Elements

3.1.4.11.2.1 GetRange

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

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

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

sheetName: The name of the sheet that contains the range. See section 2.2.5.3 for specification. MUST be present. MUST NOT be empty.

rangeCoordinates: The range to retrieve.

- MUST be present.

- MUST NOT be empty, as specified in section 2.2.4.3.

- The product of **Height** and **Width** MUST NOT be greater than 1,000,000.

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

3.1.4.11.2.2 GetRangeResponse

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

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

GetRangeResult: The returned range value.

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

3.1.4.11.3 Complex Types

None.

3.1.4.11.4 Simple Types

None.

3.1.4.11.5 Attributes

None.

3.1.4.11.6 Groups

None.

3.1.4.11.7 Attribute Groups

None.

3.1.4.12 GetRangeA1

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

```
<wsdl:operation name="GetRangeA1">
  <wsdl:input message="tns:GetRangeA1SoapIn"/>
  <wsdl:output message="tns:GetRangeA1SoapOut"/>
</wsdl:operation>
```

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

- The **rangeName** specifies the named range of the cells on the sheet to retrieve. The sheet that contains the sheet cells to retrieve is determined in the following manner:

- If **rangeName** specifies a sheet, that is the sheet which MUST be used.

- Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section 2.2.5.3. If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY<7> use any sheet in the workbook.

- Each element inside the **GetRangeA1Result** element MUST be of type **ArrayOfAnyType.** Each element inside the **GetRangeA1Result** is a row (2) of sheet cells returned from the protocol server. The number of rows (2) MUST be equal to the rows (2) in the named range specified by **rangeName**.

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

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

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

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

Subcode
DiscontiguousRangeNotSupported
EcsInvalidNamedObject
InvalidSheetName
RangeDefinitionError
RangeParseError
RangeRequestAreaExceeded
SheetRangeMismatch

3.1.4.12.1   Messages

3.1.4.12.1.1   GetRangeA1SoapIn

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

The SOAP action value is:

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

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

3.1.4.12.1.2 GetRangeA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.12.2 Elements

3.1.4.12.2.1 GetRangeA1

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

```
<xs:element name="GetRangeA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

sheetName: The name of the sheet on which the **rangeName** is defined. See section 2.2.5.3 for specification. MUST be present.

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

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

3.1.4.12.2.2 GetRangeA1Response

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

```
<xs:element name="GetRangeA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetRangeA1Result"
type="tns:ArrayOfAnyType"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
```

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

GetRangeA1Result: The returned range value.

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

3.1.4.12.3 Complex Types

None.

3.1.4.12.4 Simple Types

None.

3.1.4.12.5 Attributes

None.

3.1.4.12.6 Groups

None.

3.1.4.12.7 Attribute Groups

None.

3.1.4.13 GetSessionInformation

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

```
<wsdl:operation name="GetSessionInformation">
  <wsdl:input message="tns:GetSessionInformationSoapIn"/>
  <wsdl:output message="tns:GetSessionInformationSoapOut"/>
</wsdl:operation>
```

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

▪ The protocol server MUST return the UI culture and the data culture for the specified session (2). These are the languages specified in the OpenWorkbook operation or the languages that the protocol server fell back to, as specified in OpenWorkbook.

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

3.1.4.13.1 Messages

3.1.4.13.1.1 GetSessionInformationSoapIn

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

The SOAP action value is:

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

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

3.1.4.13.1.2 GetSessionInformationSoapOut

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

The SOAP action value is:

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

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

3.1.4.13.2 Elements

3.1.4.13.2.1 GetSessionInformation

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

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

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

3.1.4.13.2.2 GetSessionInformationResponse

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

```
<xs:element name="GetSessionInformationResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="serverVersion" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureNameUsed" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureNameUsed" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
```

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

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

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

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

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

3.1.4.13.3 Complex Types

None.

3.1.4.13.4 Simple Types

None.

3.1.4.13.5 Attributes

None.

3.1.4.13.6 Groups

None.

3.1.4.13.7 Attribute Groups

None.

3.1.4.14 GetSheetNames

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

```
<wsdl:operation name="GetSheetNames">
  <wsdl:input message="tns:GetSheetNamesSoapIn"/>
  <wsdl:output message="tns:GetSheetNamesSoapOut"/>
</wsdl:operation>
```

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

- If the session (2) is a view session in view-only mode, the number of elements inside **GetSheetNamesResult** MUST be equal to the number of visible **published sheets** in the workbook.

- If the session (2) is not a view session in view-only mode, the number of elements inside **GetSheetNamesResult** MUST be equal to the number of loaded sheets in the workbook.

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

3.1.4.14.1 Messages

3.1.4.14.1.1 GetSheetNamesSoapIn

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

The SOAP action value is:

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

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

3.1.4.14.1.2 GetSheetNamesSoapOut

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

The SOAP action value is:

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

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

3.1.4.14.2 Elements

3.1.4.14.2.1 GetSheetNames

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

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

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

3.1.4.14.2.2 GetSheetNamesResponse

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

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

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

GetSheetNamesResult: An array of **SheetInfo** elements as specified in section 3.1.4.14.3.1. MUST be present.

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

3.1.4.14.3 Complex Types

3.1.4.14.3.1 ArrayOfSheetInfo

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

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

SheetInfo: A **SheetInfo** as specified in section 3.1.4.14.3.2.

3.1.4.14.3.2 SheetInfo

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

```
<xs:complexType name="SheetInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="tns:SheetType"/>
    <xs:element minOccurs="1" maxOccurs="1" name="Visibility" type="tns:SheetVisibility"/>
  </xs:sequence>
</xs:complexType>
```

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

Type: The type of the sheet as specified in section 3.1.4.14.4.1.

Visibility: The visible condition of the sheet as specified in section 3.1.4.14.4.2.

3.1.4.14.4 Simple Types

3.1.4.14.4.1 SheetType

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

```
<xs:simpleType name="SheetType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Worksheet"/>
    <xs:enumeration value="Chartsheet"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for SheetType:

Value	Meaning
Worksheet	A worksheet.
Chartsheet	A **chart sheet**.

3.1.4.14.4.2 SheetVisibility

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

```
<xs:simpleType name="SheetVisibility">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Visible"/>
    <xs:enumeration value="Hidden"/>
    <xs:enumeration value="VeryHidden"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for SheetVisibility:

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

3.1.4.14.5 Attributes

None.

3.1.4.14.6 Groups

None.

3.1.4.14.7 Attribute Groups

None.

3.1.4.15 GetWorkbook

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

```
<wsdl:operation name="GetWorkbook">
  <wsdl:input message="tns:GetWorkbookSoapIn"/>
  <wsdl:output message="tns:GetWorkbookSoapOut"/>
</wsdl:operation>
```

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

- If **workbookType** is **FullWorkbook** or **FullSnapshot** and the user does not have Open Item permission , the protocol server returns an SOAP fault with subcode **EcsSnapshotAccessDenied**.

- If the protocol server cannot retrieve the workbook file in the form specified by **workbookType**, the protocol server returns an SOAP fault with subcode **SaveFailed.**

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

Subcode
EcsSnapshotAccessDenied
SaveFailed

3.1.4.15.1 Messages

3.1.4.15.1.1 GetWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.15.1.2 GetWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.15.2 Elements

3.1.4.15.2.1 GetWorkbook

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

```
<xs:element name="GetWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType" type="tns:WorkbookType"/>
    </xs:sequence>
  </xs:complexType>
```

71 / 150

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

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

workbookType: The form of the requested workbook as specified in section 2.2.5.4.

3.1.4.15.2.2 GetWorkbookResponse

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

```
<xs:element name="GetWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="GetWorkbookResult"
type="xs:base64Binary"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

GetWorkbookResult: A workbook file. MUST be present.

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

3.1.4.15.3 Complex Types

None.

3.1.4.15.4 Simple Types

None.

3.1.4.15.5 Attributes

None.

3.1.4.15.6 Groups

None.

3.1.4.15.7 Attribute Groups

None.

3.1.4.16 NewWorkbook

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

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

```
      </wsdl:operation>
```

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

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

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

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

3.1.4.16.1 Messages

3.1.4.16.1.1 NewWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.16.1.2 NewWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.16.2 Elements

3.1.4.16.2.1 NewWorkbook

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

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

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

path: The **trusted location (2)** of the new workbook session. MUST be present. MUST NOT be empty. MUST NOT be longer than 4095 characters. MUST be a **UNC**, or a **URI** as specified in [RFC2396].

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

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

3.1.4.16.2.2   NewWorkbookResponse

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

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

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

status: The status information returned by the protocol server.

3.1.4.16.3   Complex Types

None.

3.1.4.16.4   Simple Types

None.

3.1.4.16.5   Attributes

None.

3.1.4.16.6   Groups

None.

3.1.4.16.7   Attribute Groups

None.

3.1.4.17 OpenWorkbook

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

```
<wsdl:operation name="OpenWorkbook">
  <wsdl:input message="tns:OpenWorkbookSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookSoapOut"/>
</wsdl:operation>
```

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

1. The workbook file, specified by **workbookPath**, is specified as follows:

    1. It SHOULD be in a format specified by [MS-XLSB], [ISO/IEC-29500-1], [MS-OFFMACRO2], or [ODF] but MAY be in any other format.

    2. If it is in the [ISO/IEC-29500-1] format:

        1. It MAY<12> contain any valid SpreadsheetML records in that format.

        2. If a **odcFile** attribute of a **connection** element ([ISO/IEC-29500-1], Part 1, Section 18.13.1) is present, it SHOULD be the path of a file in the format specified by [MS-ODCFF].

        3. If a **connection** attribute of **dbPr** element ([ISO/IEC-29500-1], Part 1, Section 18.13.3) is present, it SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].

    3. If it is in the [MS-XLSB] format:

        1. It MAY<13> contain any valid records in that format.

        2. If a **brtBeginExternalConnection** record is present and it has the **stConnectionFile** field present, the **stConnectionFile** SHOULD be the path of a file in the format specified by [MS-ODCFF].

        3. If a **BrtBeginECDbProps** record is present, its **stConn** field SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].

    4. If a file in [MS-ODCFF] is referenced by the workbook file:

        1. **SourceFile** element of **OfficeDataConnection** ([MS-ODCFF] section 2.7.1.1) MUST NOT be present.

        2. **Parameter** elements of **CT_Connection** ([MS-ODCFF] section 2.7.1.1) MUST be ignored.

2. If the language represented by **uiCultureName** is unsupported on the protocol server, the protocol server MAY<14> fall back to some other supported language. **uiCultureName** MAY<15> be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

3. If the language represented by **dataCultureName** is unsupported on the protocol server, the protocol server MAY<16> fall back to some other supported language. **dataCultureName** MAY<17> be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

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

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

Subcode
FileOpenNotFound
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError
WorkbookNotSupported

3.1.4.17.1 Messages

3.1.4.17.1.1 OpenWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.17.1.2 OpenWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.17.2 Elements

3.1.4.17.2.1 OpenWorkbook

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

```
<xs:element name="OpenWorkbook">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
```

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

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

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

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

3.1.4.17.2.2 OpenWorkbookResponse

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

```xml
<xs:element name="OpenWorkbookResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookResult" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

status: The status information returned by the protocol server.

3.1.4.17.3 Complex Types

None.

3.1.4.17.4 Simple Types

None.

3.1.4.17.5 Attributes

None.

3.1.4.17.6 Groups

None.

3.1.4.17.7 Attribute Groups

None.

3.1.4.18   OpenWorkbookForEditing

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

```
<wsdl:operation name="OpenWorkbookForEditing">
  <wsdl:input message="tns:OpenWorkbookForEditingSoapIn"/>
  <wsdl:output message="tns:OpenWorkbookForEditingSoapOut"/>
</wsdl:operation>
```

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

1. The workbook file, specified by **workbookPath**, is specified as follows:

    1. It SHOULD be in a format specified by [MS-XLSB], [ISO/IEC-29500-1], [MS-OFFMACRO2], or [ODF] but MAY be in any other format.

    2. If it is in the [ISO/IEC-29500-1] format:

        1. It MAY<18> contain any valid SpreadsheetML records in that format.

        2. If a **odcFile** attribute of a **connection** element ([ISO/IEC-29500-1], Part 1, Section 18.13.1) is present, it SHOULD be the path of a file in the format specified by [MS-ODCFF].

        3. If a **connection** attribute of **dbPr** element ([ISO/IEC-29500-1], Part 1, Section 18.13.3) is present, it SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].

    3. If it is in the [MS-XLSB] format:

        1. It MAY<19> contain any valid records in that format.

        2. If a **brtBeginExternalConnection** record is present and it has the **stConnectionFile** field present, the **stConnectionFile** SHOULD be the path of a file in the format specified by [MS-ODCFF].

        3. If a **BrtBeginECDbProps** record is present, its **stConn** field SHOULD be in the format specified by [MS-ODBCSTR] or [MS-OLEDBSTR].

    4. If a file in [MS-ODCFF] is referenced by the workbook file:

        1. **SourceFile** element of **OfficeDataConnection** ([MS-ODCFF] section 2.7.1.1) MUST NOT be present.

        2. **Parameter** elements of **CT_Connection** ([MS-ODCFF] section 2.2.1) MUST be ignored.

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

3. If the language represented by **dataCultureName** is unsupported on the protocol server, the protocol server MAY<22> fall back to some other supported language. **dataCultureName** MAY<23> be empty, in which case the determination of which language to default to is protocol-server-implementation specific.

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

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

Subcode
FileOpenNotFound
FileOpenAccessDenied
FileCorrupt
GenericFileOpenError

3.1.4.18.1 Messages

3.1.4.18.1.1 OpenWorkbookForEditingSoapIn

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

The SOAP action value is:

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

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

3.1.4.18.1.2 OpenWorkbookForEditingSoapOut

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

The SOAP action value is:

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

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

3.1.4.18.2 Elements

3.1.4.18.2.1 OpenWorkbookForEditing

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

```
<xs:element name="OpenWorkbookForEditing">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"/>
```

79 / 150

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

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

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

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

3.1.4.18.2.2   OpenWorkbookForEditingResponse

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

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

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

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

3.1.4.18.3   Complex Types

None.

3.1.4.18.4   Simple Types

None.

3.1.4.18.5   Attributes

None.

3.1.4.18.6   Groups

None.

3.1.4.18.7   Attribute Groups

None.

3.1.4.19 Refresh

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

```
<wsdl:operation name="Refresh">
  <wsdl:input message="tns:RefreshSoapIn"/>
  <wsdl:output message="tns:RefreshSoapOut"/>
</wsdl:operation>
```

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

1. If **connectionName** is not present or is empty, the protocol server refreshes all **data connections (2)** in the workbook if any exist. If **connectionName** is present and is not empty, the protocol server refreshes the specified data connections (2) in the workbook.

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

3.1.4.19.1 Messages

3.1.4.19.1.1 RefreshSoapIn

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

The SOAP action value is:

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

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

3.1.4.19.1.2 RefreshSoapOut

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

The SOAP action value is:

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

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

3.1.4.19.2 Elements

3.1.4.19.2.1 Refresh

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

```
<xs:element name="Refresh">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string"/>
```

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

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

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

3.1.4.19.2.2 RefreshResponse

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

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

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

3.1.4.19.3 Complex Types

None.

3.1.4.19.4 Simple Types

None.

3.1.4.19.5 Attributes

None.

3.1.4.19.6 Groups

None.

3.1.4.19.7 Attribute Groups

None.

3.1.4.20 SaveWorkbook

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

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

```
</wsdl:operation>
```

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

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

Subcode
FileSaveGeneric
OperationNotSupportedInViewOnly
SaveFailed
SavingUnsupportedInViewMode
SavingWorkbookAccessDenied
SavingWorkbookLockedOrCheckedOut
SavingWorkbookNoWorkbookName

3.1.4.20.1 Messages

3.1.4.20.1.1 SaveWorkbookSoapIn

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

The SOAP action value is:

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

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

3.1.4.20.1.2 SaveWorkbookSoapOut

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

The SOAP action value is:

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

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

3.1.4.20.2 Elements

3.1.4.20.2.1 SaveWorkbook

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

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

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

3.1.4.20.2.2 SaveWorkbookResponse

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

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

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

3.1.4.20.3 Complex Types

None.

3.1.4.20.4 Simple Types

None.

3.1.4.20.5 Attributes

None.

3.1.4.20.6 Groups

None.

3.1.4.20.7 Attribute Groups

None.

3.1.4.21 SaveWorkbookCopy

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

```
<wsdl:operation name="SaveWorkbookCopy">
  <wsdl:input message="tns:SaveWorkbookCopySoapIn"/>
```

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

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

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

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

Subcode
EcsSnapshotAccessDenied
FileSaveGeneric
SaveFailed
SavingWorkbookAccessDenied
SavingWorkbookFileNameExists
SavingWorkbookInvalidFileName
SavingWorkbookInvalidLocation

3.1.4.21.1 Messages

3.1.4.21.1.1 SaveWorkbookCopySoapIn

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

The SOAP action value is:

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

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

3.1.4.21.1.2 SaveWorkbookCopySoapOut

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

The SOAP action value is:

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

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

3.1.4.21.2 Elements

3.1.4.21.2.1 SaveWorkbookCopy

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

```
<xs:element name="SaveWorkbookCopy">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="workbookType" type="tns:WorkbookType"/>
      <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

▪ MUST be present.

   If the specified value is not a Uniform Resource Identifier (URI) as specified in [RFC2396], or a Universal Naming Convention (UNC), the protocol server MUST return a SOAP fault response with subcode **SavingWorkbookInvalidFileName**.

▪ The protocol server ignores the number sign ("#") and fragment, as specified in [RFC3986].

workbookType: A **WorkbookType** as specified in section 2.2.5.4.

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

3.1.4.21.2.2 SaveWorkbookCopyResponse

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

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

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

3.1.4.21.3 Complex Types

None.

3.1.4.21.4 Simple Types

3.1.4.21.4.1 SaveOptions

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

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

The following table specifies the allowable values for SaveOptions:

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

3.1.4.21.5 Attributes

None.

3.1.4.21.6 Groups

None.

3.1.4.21.7 Attribute Groups

None.

3.1.4.22 SetCalculationOptions

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

```
<wsdl:operation name="SetCalculationOptions">
  <wsdl:input message="tns:SetCalculationOptionsSoapIn"/>
  <wsdl:output message="tns:SetCalculationOptionsSoapOut"/>
</wsdl:operation>
```

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

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

3.1.4.22.1 Messages

3.1.4.22.1.1 SetCalculationOptionsSoapIn

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

The SOAP action value is:

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

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

3.1.4.22.1.2 SetCalculationOptionsSoapOut

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

The SOAP action value is:

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

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

3.1.4.22.2 Elements

3.1.4.22.2.1 SetCalculationOptions

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

```
<xs:element name="SetCalculationOptions">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="calcType"
type="tns:WorkbookCalculation"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

3.1.4.22.2.2 SetCalculationOptionsResponse

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

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

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

3.1.4.22.3 Complex Types

None.

3.1.4.22.4 Simple Types

3.1.4.22.4.1 WorkbookCalculation

The **WorkBookCalculation** type is an enumeration of a set of options that specify how the protocol server calculates the workbook. This type is defined as follows.

```
<xs:simpleType name="WorkbookCalculation">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Manual"/>
    <xs:enumeration value="Automatic"/>
    <xs:enumeration value="Semiautomatic"/>
  </xs:restriction>
</xs:simpleType>
```

The following table specifies the allowable values for WorkbookCalculation:

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

3.1.4.22.5 Attributes

None.

3.1.4.22.6 Groups

None.

3.1.4.22.7 Attribute Groups

None.

3.1.4.23   SetCell

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

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

1. The row (2) and column (2) fields on the sheets specified by **sheetName** specify the location of the worksheet cell to set.

2. **cellValue** MUST either be empty and have the **xsi:nil** attribute set to **true** or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string,** or **unsigned byte.**

   1. If the value is of type **string**, it MUST NOT be longer than 32,767 characters.

   2. If the protocol server does not support the value it MUST return a SOAP fault.

   3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

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

Subcode
FormulaParseFailed
InvalidSheetName
CannotEditPivotTable

3.1.4.23.1   Messages

3.1.4.23.1.1   SetCellSoapIn

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

The SOAP action value is:

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

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

3.1.4.23.1.2   SetCellSoapOut

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

The SOAP action value is:

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

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

3.1.4.23.2 Elements

3.1.4.23.2.1 SetCell

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

```
<xs:element name="SetCell">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int"/>
      <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int"/>
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

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

column: The column (2) index of the worksheet cell. MUST be greater than or equal to 0 and less than or equal to 16,383.

cellValue: The value to set in the worksheet cell.

3.1.4.23.2.2 SetCellResponse

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

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

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

3.1.4.23.3 Complex Types

None.

3.1.4.23.4 Simple Types

None.

3.1.4.23.5 Attributes

None.

3.1.4.23.6 Groups

None.

3.1.4.23.7 Attribute Groups

None.

3.1.4.24 SetCellA1

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

```
<wsdl:operation name="SetCellA1">
  <wsdl:input message="tns:SetCellA1SoapIn"/>
  <wsdl:output message="tns:SetCellA1SoapOut"/>
</wsdl:operation>
```

The protocol client sends a **SetCellA1SoapIn** request message, and the protocol server responds with a **SetCellA1SoapOut** response message, as follows:

1. The sheet that contains the worksheet cell that will be set is determined in the following manner:

   ▪If **rangeName** specifies a sheet, that sheet MUST be used.

   ▪Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section 2.2.5.3.

   ▪If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY<24> use any sheet in the workbook.

2. If **sheetName** is empty and **rangeName** specifies a sheet, as defined in section 2.2.5.2, the protocol server MUST use the sheet that is specified by **rangeName** to locate the range.

3. **cellValue** MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string,** or **unsigned byte.**

   1. If the value is of type **string**, it MUST NOT be longer than 32767 characters.

   2. If the protocol server does not support the value, it MUST return a SOAP fault.

   3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

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

Subcode
FormulaParseFailed
InvalidSheetName
DimensionAndArrayMismatch
SheetRangeMismatch
RangeParseError

3.1.4.24.1 Messages

3.1.4.24.1.1 SetCellA1SoapIn

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

The SOAP action value is:

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

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

3.1.4.24.1.2 SetCellA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.24.2 Elements

3.1.4.24.2.1 SetCellA1

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

```
<xs:element name="SetCellA1">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="cellValue"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

rangeName: The range of the worksheet cell to set. See section 2.2.5.2. MUST be present.

cellValue: The value to set in the worksheet cell.

3.1.4.24.2.2 SetCellA1Response

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

```
<xs:element name="SetCellA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

3.1.4.24.3 Complex Types

None.

3.1.4.24.4 Simple Types

None.

3.1.4.24.5 Attributes

None.

3.1.4.24.6 Groups

None.

3.1.4.24.7 Attribute Groups

None.

3.1.4.25 SetParameters

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

```
<wsdl:operation name="SetParameters">
  <wsdl:input message="tns:SetParametersSoapIn"/>
  <wsdl:output message="tns:SetParametersSoapOut"/>
</wsdl:operation>
```

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

- For each ParameterInfo (section 3.1.4.25.3.2) in **parameters**, if the workbook parameter is a PivotTable page field:

  - If the number of elements in the **Values** array of ParameterInfo is equal to or greater than 1, the protocol server uses the values as the PivotTable **manual filter**.

  - If the number of element in the **Values** array of ParameterInfo is 0, the protocol server changes the PivotTable manual filter to the **default filter value**.

- For each ParameterInfo in **parameters**, if the workbook parameter is a **slicer** page filter:

  - If the number of elements in the **Values** array of ParameterInfo is equal to or greater than 1, the protocol server uses the values as the slicer manual filter.

  - If the number of elements in the **Values** array of ParameterInfo is 0, the protocol server changes the slicer manual filter to the default filter value.

- For each ParameterInfo in **parameters**, if the workbook parameter is neither a slicer page filter nor a PivotTable page filter:

  - If the number of elements in the **Values** array of ParameterInfo is not exactly 1, the protocol server ignores all elements in the array.

  - If the number of elements in the **Values** array of ParameterInfo is exactly 1, the protocol server applies that value to the parameter.

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

Subcode
FormulaEditingNotEnabled
InvalidSheetName
ParameterDoesNotExist
SetParameterFailure

3.1.4.25.1 Messages

3.1.4.25.1.1 SetParametersSoapIn

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

The SOAP action value is:

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

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

3.1.4.25.1.2 SetParametersSoapOut

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

The SOAP action value is:

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

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

3.1.4.25.2 Elements

3.1.4.25.2.1 SetParameters

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

```
<xs:element name="SetParameters">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="1" name="parameters"
type="tns:ArrayOfParameterInfo"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

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

3.1.4.25.2.2 SetParametersResponse

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

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

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

3.1.4.25.3 Complex Types

3.1.4.25.3.1 ArrayOfParameterInfo

An array of **ParameterInfo** elements defined as follows. The number of element in the **ArrayOfParameterInfo** element MUST be less than or equal to 255.

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

ParameterInfo: A workbook parameter as specified in section 3.1.4.25.3.2.

3.1.4.25.3.2   ParameterInfo

Represents a value or a list of values to be set into a workbook parameter in the workbook and is defined as follows.

```
<xs:complexType name="ParameterInfo">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="Values" type="tns:ArrayOfString"/>
  </xs:sequence>
</xs:complexType>
```

Name: The name of the workbook parameter. MUST NOT exceed 255 characters.

Values: An array of string values to set in the workbook parameter as specified in section 3.1.4.25.3.3.

3.1.4.25.3.3   ArrayOfString

An array of string values defined as follows.

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

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

3.1.4.25.4   Simple Types

None.

3.1.4.25.5   Attributes

None.

3.1.4.25.6   Groups

None.

3.1.4.25.7   Attribute Groups

None.

3.1.4.26   SetRange

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

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

1. The **rangeCoordinates** on the sheet specified by **sheetName**; specify the location of the worksheet cells to set.

2. If the user does not have the Open Item permission on the workbook, **rangeValues** SHOULD<25> contain only one item.

3. Each element inside the **rangeValues** element MUST be of type **ArrayOfAnyType** and be a row (2) of worksheet cells. The number of rows (2) in **rangeValues** MUST be equal to the number of rows (2) specified in the **rangeCoordinates** element.

4. Each element inside each of the **ArrayOfAnyType** type nested inside **rangeValues** MUST be a single worksheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of columns (2) specified in the **rangeCoordinates** element.

5. All worksheet cells in the **rangeValues** element MUST either be empty and have the xsi:nil attribute set to true or one be of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string, or unsigned byte.**

    1. If the value is of type **string**, it MUST NOT be longer than 32,767 characters.

    2. If the protocol server does not support the value, it MUST return a SOAP fault.

    3. If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

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

Subcode
FormulaParseFailed
InvalidSheetName
DimensionAndArrayMismatch
SheetRangeMismatch
RangeParseError

3.1.4.26.1 Messages

3.1.4.26.1.1 SetRangeSoapIn

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

The SOAP action value is:

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

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

3.1.4.26.1.2 SetRangeSoapOut

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

The SOAP action value is:

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

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

3.1.4.26.2 Elements

3.1.4.26.2.1 SetRange

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

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

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

sheetName: The name of the sheet that contains the range to set. See section 2.2.5.3. MUST be present. MUST NOT be empty.

rangeCoordinates: The range to set. See section 2.2.5.2. MUST be present. MUST NOT be empty.

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

3.1.4.26.2.2 SetRangeResponse

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

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

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

3.1.4.26.3 Complex Types

None.

3.1.4.26.4 Simple Types

None.

3.1.4.26.5 Attributes

None.

3.1.4.26.6 Groups

None.

3.1.4.26.7 Attribute Groups

None.

3.1.4.27 SetRangeA1

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

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

1. The sheet that contains the range that will be set is determined in the following manner:

   ▪If **rangeName** specifies a sheet, that sheet MUST be used.

   ▪Otherwise, the sheet specified in **sheetName** MUST be used, as specified in section 2.2.5.3.

   ▪If neither **sheetName** nor **rangeName** specify a sheet, the protocol server MAY<26> use any sheet in the workbook.

---

2. If **sheetName** is empty and **rangeName** specifies a sheet, as defined in section 2.2.5.2, the sheet that is specified by **rangeName** MUST be used to locate the range.

3. Each element inside the **rangeValues** element MUST be of type **ArrayOfAnyType** and be a row (2) of worksheet cells. The number of rows (2) in **rangeValues** MUST be equal to the number of rows (2) specified in the **rangeName** element.

4. Each element inside each of the **ArrayOfAnyType** type nested inside **rangeValues** MUST be a single worksheet cell. The number of elements inside each such nested **ArrayOfAnyType** MUST be equal to the number of columns (2) specified in the **rangeName** element.

5. All worksheet cells in the **rangeValues** element MUST either be empty and have the xsi:nil attribute set to true or be one of the following types: **boolean**, **dateTime**, **decimal**, **double**, **float**, **int**, **short**, **string,** or **unsigned byte.**

   ▪If the value is of type **string**, it MUST NOT be longer than 32767 characters.

   ▪If the protocol server does not support the value, it MUST return a SOAP fault.

   ▪If the protocol server does not support setting a value in the specified location, it MUST return a SOAP fault.

6. If the user does not have the Open Item permission on the workbook, **rangeValues** SHOULD<27> contain only one worksheet cell.

3.1.4.27.1   Messages

3.1.4.27.1.1   SetRangeA1SoapIn

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

The SOAP action value is:

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

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

3.1.4.27.1.2   SetRangeA1SoapOut

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

The SOAP action value is:

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

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

3.1.4.27.2   Elements

3.1.4.27.2.1   SetRangeA1

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

```
<xs:element name="SetRangeA1">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string"/>
    <xs:element minOccurs="0" maxOccurs="1" name="rangeValues" type="tns:ArrayOfAnyType"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

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

sheetName: The name of the sheet that contains the range to set. See section 2.2.5.3. MUST be present.

rangeName: The name of the range to set. See section 2.2.5.2. MUST be present.

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

3.1.4.27.2.2   SetRangeA1Response

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

```
<xs:element name="SetRangeA1Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

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

3.1.4.27.3   Complex Types

None.

3.1.4.27.4   Simple Types

None.

3.1.4.27.5   Attributes

None.

3.1.4.27.6   Groups

None.

3.1.4.27.7   Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4   Protocol Examples

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

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

4.1   Getting the Value of a Worksheet Cell

This example shows a simple, and typical, flow of how this protocol is used to enable a protocol client to get calculated results from a workbook. This example assumes a workbook has already been created and is stored in a location that the protocol server can load workbook files from.

In this example, a workbook is loaded on the protocol server, recalculated, a value is fetched, and the workbook is closed. The following UML diagram illustrates this example.
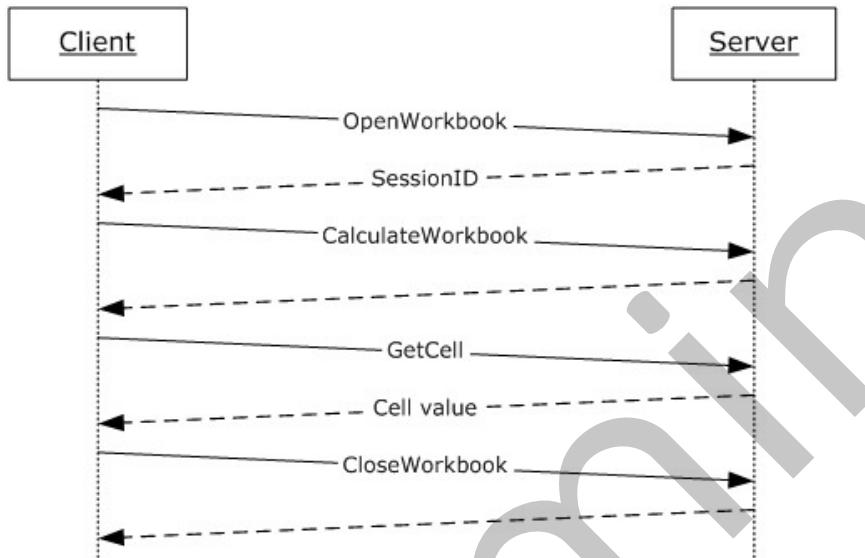
Figure 3: Getting the value of a worksheet cell example

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

▪   Load the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <workbookPath>http://server/site/documents/book2.xlsx</workbookPath>
      <uiCultureName />
      <dataCultureName />
    </OpenWorkbook>
  </soap:Body>
```

```
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <OpenWorkbookResult>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</OpenWorkbookResult>
    </OpenWorkbookResponse>
  </soap:Body>
</soap:Envelope>
```

▪ Recalculate the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <calculateType>CalculateFull</calculateType>
    </CalculateWorkbook>
  </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

▪ Fetch the specific value from a worksheet cell within the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <GetCellA1 xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
       <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
       <sheetName>Sheet1</sheetName>
       <rangeName>A1</rangeName>
       <formatted>false</formatted>
     </GetCellA1>
   </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <GetCellA1Response xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
       <GetCellA1Result xsi:type="xsd:double">5</GetCellA1Result>
     </GetCellA1Response>
   </soap:Body>
</soap:Envelope>
```

▪ Close the workbook session on the protocol server.

Request

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <CloseWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
       <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.5791QXImNo/JutQkV4kbZDd90.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
     </CloseWorkbook>
   </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
   </soap:Body>
</soap:Envelope>
```

4.2 Refreshing Data and Retrieving a Workbook

This example illustrates how this protocol is used to enable a more complex series of actions, including refreshing connected data and retrieving the resulting workbook after a calculation. This example assumes a workbook has already been created and is stored in a location that the workbook protocol server can load files from. In this example a workbook is loaded on the protocol server, some values are set into the workbook, the connected data is refreshed, the workbook is recalculated, and finally the desired result of that calculation is retrieved and the workbook is closed. The following UML diagram illustrates this example.

Figure 4: Refresh data and retrieve workbook example

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

▪ Load the workbook.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <OpenWorkbook xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
      <workbookPath>http://server/site/documents/book2.xlsx</workbookPath>
      <uiCultureName />
      <dataCultureName />
```

```
      </OpenWorkbook>
   </soap:Body>
</soap:Envelope>
```

Response

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

▪ Set the value of the worksheet cell.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <SetCellA1 xmlns="http://schemas.microsoft.com/office/excel/server/webservices">
       <sessionId>36.c22a23f3-976c-4933-aa85-
74c6f06c51b5164.1.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000#+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
       <sheetName>Sheet1</sheetName>
       <rangeName>A1</rangeName>
       <cellValue xsi:type="xsd:int">5</cellValue>
     </SetCellA1>
   </soap:Body>
</soap:Envelope>
```

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <soap:Body>
     <SetCellA1Response xmlns="http://schemas.microsoft.com/office/excel/server/webservices"
/>
   </soap:Body>
</soap:Envelope>
```

▪ Refresh the connected data in the workbook.

Request

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

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <RefreshResponse xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
</soap:Envelope>
```

▪ Recalculate the workbook.

Request

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

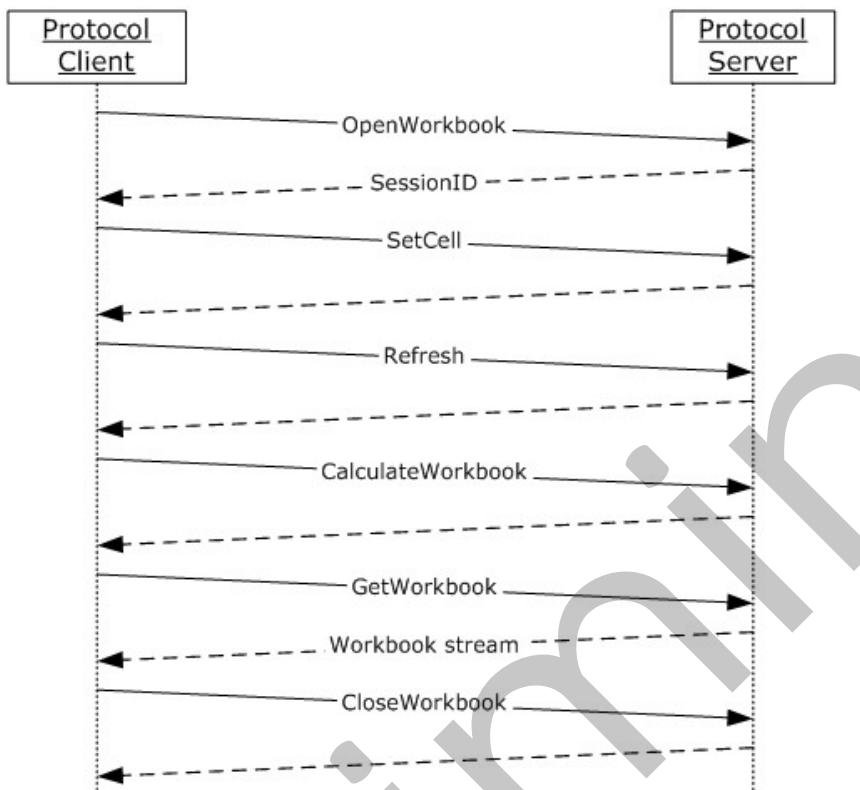**Response**

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CalculateWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
```

```
</soap:Envelope>
```

▪ 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.V23.494r81aHjSSiziqJhYlU2B590.5.en-US5.en-US73.+0480#0000-11-00-
01T02:00:00:0000##+0000#0000-03-00-02T02:00:00:0000#-006036.4d387825-9c48-4d48-98fe-
7894661941631.N</sessionId>
      <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>UEsDBBQABgAIAAAAIQAezCq/7QEAAFYKAAATAAgCW0NvbnRlbnRfVHlwZXNdLnhtbCCiBAIooA
ACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADMlk2P0zAQhu9I/IfIV9S4uyCEUNM97MIRVtoicXWdaWPVX7Knpf33jJ3dsq
q6DSGR2EuSNvb7PjO2ZzK72Rtd7CBE5WzFrsopK8BKVyu7rtiPxdfJJ1ZEFLYW2lmo2AEiu5m/fTNbHDzEgmbbbWLEG0X/
mPMoGjIil82DpzcoFI5B+hjX3Qm7EGvj1dPqRS2cRLE4wabD57A5WYqux+LKnv1uSpbKsuG3HJauKCe+1kgIJlO9sfWIy
cauVklA7uTUkXUYfQNSxAUCjSx8UOYYYHQKTAIuNnPQPo2M/0MaqSZmaw2Cgf31HoLzikNy9H9TjvOy1HUDUU9yLgN2Eod
r7X/JcLm6Vzm/KySN/U5BSVRij7xH3BPw+OPN+uRgZJ8WXhhzXr4Tj/X/i8Grn8FbIBvifRzpRyqp0VsZepnMeHSuGVB
WA5+twmizTYUj1xYJM0ceRF+WZcgdDxIOGse1b0S7nRgSoH5Bq3np0gOfaHRx5pyzEktLQ7sz8PHwHnJT2tP2z8iUc6gv
3wflIrsdA/z3xVOfT7IknIQio4Fjpz1XMoyP1rf6GJ90MUmOsoe7rLbcRnRls38r8pXk7+KfRXCGYnPUPIxGQaNLrTv4Z
huHl+Sj6zwzDW9VwhuHn7wIDz1+F898AAAD//wMAUEsDBBQABgAIAAAAIQATXr5lBQEAAN8CAAALAAgCX3JlbHMvLnJlb
HMgogQCKKAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAArJLPTsMwDMbvSLxDlPvqqbiCE0NJdJqTdTdECoPYBL3j9rGUZJB9
/YEJASVRrsDx9ifP//8KdvdOPTijXxxo2Sq5znIpyGo2ra2VfCkfV//dShIjWYM+WlDxRkLvi+mr7TD3GNBSa1gWRXGGxQso
nRPQAE3dCAIWNHNnUq9gPG9gPQ1ONQd1gSbPL8D/9tDFhNPcTBK+oO5kaI8ubR52ZurqtW0Z30Z30cyMYzK4DGSNaQWTmf2Hx
s0zWiRF9TVNKKwfkrlAOhclrAlnCfaXIE7097UwUESDEUGzp3meT8Uc0PpyoOWIpoqfdMYe3tl3r8zdHMvtf7LoY4g8LITz
pflGgsm3LD4AAAD//wMAUEsDBBQABgAIAAAAIQDo36Q+cAEAAD8HAAAaAAgBeGwvX3JlbHMvd29ya2Jvb2sueG1sLnJlb
HMgogQBKKAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

[MS-ESP] — v20120410
Excel Web Services Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: Tuesday, April 10, 2012

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACslU1PwzAMhu9I/Icqd
5p1G+NDa3dgQtoVhsQ1St02WptUSTbov8dUKNnECDvkUjV2az957TjL1WfXJgfQRiiZkyydkAQkV6WQdU7ets839yQxls
mStUpCTgYwZFVcXy1foGUWfzKN6E2CUaTJSWNt/0ip4Q1OzKSqB4meSumOWVzqvmvaM71gNdDqZLKg+jkGGGk5jJpsyJ3pS
Yfzv0mPn/2KqqBIe14vsOpD2TgpqGaShfrcbtGQzMdA02JyfmFIkjPQ8ziwnzofTONADWgziTQVT0zEIWdzFhjB1aLK2X
ZFyH0mfTmPn53ljVvaP0DiFNqbNSYaGbh3Ci0rg6eBpn+iNNNASTxZTGZf4bJGvBLGLCcCU18PHge5wjYwgkiyqL6w3P8
atjgufnNqYuFiceeJJxScdnsDYzztt44+0SSYJ9O49J04uDsk8M74I1VEKK76bxCnkv9a/+w6BqDzExLxHN0dCTa6/4Ag
AA//8DAFBLAwQUAAYACAAAACEAwIM9PoUBAADFAgAADwAAAHsL3/dvcmtib29rLlnhtbIxSy27DIBC8V+o/IO4Nju08GsW
J1JfaSxWpaXOmZh2jYLCAxMnfdyHKQ2oPPbGzzA6zI6bzfaPAPIDqyTRhe030soAV0aIfW6oJ/Ll7sxJc5zLbgyVGgp6AEfn
s9ubaWfs5tuYDUEB7Qpae99OGHNlDQ13PdOCxpvK2IZ7HbNXSGuBC1cD+EaxNEmGrOFS06PCxP5Hw1SVLOHJlNsGtD+KW
FDco31Xy9bR2bSSCr6OGxHetu+8Qd97RYnijz8L6UEUdIDQdHBp5JTYbfuwlQpvx8NkSNnsvOTCE1lT1YBdW7nh5wKQoEV
DxrfJLXPj0HvbTPE3jbAjnS0LnLjIBkv1KamG6QMWwDyeUIeijzUoKXxc0Te8zdHnsvYJc1x5nBsloEJyxK/kYKT4TT6L
jvh8hZjQZe2+4EtZ2IrGwb6IfFH6x0ys21md2+ic7u2JjfWZn0V0UR0slVyWGF45oIs+zfh4YrdwZ/8jLGpB1AcjEVqCO
89GVapxhFx4OsdMPnP0AAAD//wMAUEsDBBQABgAIAAAAIQAllvEaQEAAIwCAAAYAAAAEgwvd29ya3NoZWV0cy9zaGVld
DIueG1sjJJNT8MwDIbvSPyHKKHeajm9N6xASQuyAhPi6Z6nbRmviyPEY/Hu8jCGkXbjZsfPEr9/Mbj7DqD6AssfY6ElVaw
XRYetj3+i31/uTa60y29jaESM0+guyvpkfH802SKs8ALASQsyNHpjT1JjsBgg2V5ggSqVDCpYlpd7kRGDbcimM5rSuL02
wPuodYUr/YWDXeQd36NYBIu8gBKNlmT8PPuU9Lbj/4IKl1TqdOOAxJEEs/ev4qUK2Cmy76iGSXo+j+nJxbt2eX5AAfvCPM
2HElOLMb9FDztTklQprPWi8KtmtXBF2jbyfazGdlOe8eNvlPrLa7XiKutoVFz2+h622oOeu/Lrp9ItdDZ9cjPuHkA3w8sx
l780u8sW7mebA+Plnofsxqhk566utKKdv0lZkzl9EKrJTJj2GeD2AhiVl2dadUh8j754b4Ar5NC8mJRcabRCYnJepYXpl
4k0KItgs3vP5p/AwAA//8DAFBLAwQUAAYACAAAACEAN4tbid4AAADZAQAAIwAAAHsL3/dvcmtzaGVldHMvX3JlbHMvc2h
lZXQxLnhtbC5yZWxzxrJHBasMwDIbvg76Db120sMYo04vY9Drmj2A6yiJaSIbSyvt28+7lKUUdtlN0o8+/ZK2u8s8qTNm
DpEs1LoCheRjF2iw8Nm+r19AsJqjq3BQJLVyRYdesnrYfODpTYGxGx1KpQiC2MIunVGPYjo51+F9lVyBbpDVBYNYVk65M9
uHbT2h2TbBM35Nm+r19AsXQ5j/BQY7DdXdt+Smgrr28M4ehy9DcOmy60Lrp9iR6gmAiFdrRBBs6BZhESHF6WOenZQ09DyY5f5IDm
k1VPZv8mwHNggn2nYW87zag2msqk/9mx74PHt+i/5qR5MEIk3IgwXxXxAkbIgIgF7TLA4oFre1+1RiQetH9ph+3/zWWx0BgHpus/9VkOEdp3X
HCpb9bmU26xbUuB/6xZRYPab4BAAD//wMAUEsDBBQABgAIAAAAIQATxCwTwgAAAEIBAAAjAAAAeGwvd29ya3NoZWV0cy9
fcmVlscy9zaGVldDIuLnJlbHHOEj8FqwzAQRO+F/IPY7Q7h1CKJ4Xdcm/QBFXtui9kPdDIB/W89Imcf1iyhZkEKOw2PeMF1/
WRf1i4VjIgutbkAhhhYTREmix8nT62r6BYPA1+SYGWrsjQu81UJsYiuYFnbJe8BzwmmHH1rFGGGqmRMZfVSY5M9
uHbT2h2TbMM35a8DsD3J1THQYL5TT00oE7XXXXXXXXXXXXXXXX

L5+jNy+fHT9+cfz45+MnT44f/2R1ORN3cRzkJ77+7vO/vvkE/fn829dPvyyWl3n533789NdfvigWBI41W9Grr579/uLZq
68/++P7pwXi20Bn8uI9GhGJ7pAjdMAj8M0A466c9MX5ZvRCTJ0ZOATdBarbKnQE70wwK5JrEhe8+wLoZZHgjfFDZ63dUI
wVLbB8K4wcwT3OWROoXhEAt7StHMK9cRwUGxfjvNwBxodFtls4dkLbHifAq7OkdLBvhcRZ5j7DscIBiYlC+h4fEVLg3QN
KHVz3Mn6KHlDUxLQQkh7tO4k0m7RLI4jLpMhnCLWDzd591OSsyOsdcuhKwobArGDxPcIcGG/gscJRkcoejlge8NtYhUWL
7E6En5drSwWRDgjjqD0gUhbNuSvA31zQb2EopoVh32OTyJUUio6KdN7GnOcld/ioFWJ48ChAoUvjMC97U44gRTHa56pIf
I+7O0RfQxxwvDDc9ylxwn16IbhHA2dJswTRd8ZCewGF2qm/EYUHqM7CYswoVGPr/WUxbnjb0JqKtsTuiRK8SG4JC+8OHs
f7BHJ9vvFc1t3Luuv95+vuor181mo7K7BQezV5sLzYsORoIUkeUsa6asLIbW14soRmMejAoJ5njhDJ9FgtCeFjWtwduUB
gMwcJrj6mKuyGOAGOXfG0kkCmqgOJEi7h9M8MF+rW8sDTlT07rOmnLVsPJFZ7fGCHV/WwGQdHp2pMyvnMCWVmaFUrOKux
1fVUKeh8G2MVvagzW6uYpZlS51ibulzoGgxO0QQWgoC7AMprcLCoTcOzCWZkoHG3DTgLi4nCRYZIhnhA0hhpv+djVDFBy
nLFHBdD7hTESD+EnoJazlpdq30Ha2cJUt5cdYG5LHrvEqUsg2dRAm0ntyOL85uTxeio4dVrKzUP+ThpeEN4rIWPUQJRl5
r4YRbA2wNfCZv2p25ms8tn0axnjrmboAKHnhb3OYedOpAIgXawDG1qmFtpCrBYW7LrX6kBrBflgM30t1jF6gYkw7+2CsD
RDS0ZDomv8sHOjWjs7GVaSvlYEdENB0eoz8biAEP4daqCPwMq4XTCVAR9AS9bNNrmlluc002XP6g3cqDjfT4Q0lBgloQ4
7QW6fmRlxvpiiswUIHOVww6ALwTWIH9+nE09+j/gnC8Alzhf3GvAk/l8Xpw1A4JToNWBVuTDK1yBkS7DDY8LFXJoLklI/
Y4APmhaAhQBeA8Lt2Gfw4tk81+QQ/3fllKrQ2tj8DCvDmiABAWaoUJByD50G1NUTlFWSSmJVZkpMnsxt1yZ2GX3ySFhPd
3a1jRl81AIFcw0ibS6G7mTO9e9TgtjP9DcNV9GndYwZYu2evzThNbWaHDKba+Gp2b4T5dYQJbsfDM9o1R5R/SNGXuuZqk
ExnIdvp4WzLdcwjkZlG1Ecx6v1LLFQRTnPYbBKc+Ft/Eh0n+A11DhM/utBM2TevwAWiaCLxloZZA2kNVXLJ9Euu/ZwT7w
YTtok0mrstCmjFijlnGwC36Amdo9AbZe2VnifU6wp5zbNefsxYsEO0XYwdqOLYQaIntyi8LQMHs+NYExXtf2fJf+OE9x9Co
Hfgzc2YKWmSCb5PIjA8UXTNPoDNby2aqVt/AwAA//8DAFBLAwQUAAYACAAAACEs6QaM6YCAACYBgAAAAAHhsL3dvcm
tzaGVldsHMvc2hlZXXQxLnhtbIyVS2/2iMBCA7yvtf7B8Bl0l4NAERqqqElsNKq+0+zsaZEKtJnLUNtP9+xzYJDQ+pB8Aez3z
zsGeYP71VJTmA0kLWKY7GISVj1+hDkxU7eaS4J6Gd6G07H9WsD/WF/n9TBNbWaHDKba+Gp2b4T5dYQJbsfDM9o1R5R/SNG
NZ7kUlXM4FbtAt0oYjzqzpqspgGIAPQcVETT1hpj7DkHkuOKwk31dQGw9RUDKD8etCNLqlVfwzuIqp133zwGXVIGIrSmHeH
ZSSis82u1oqti0x77dozHjLdpsrfCW4klrmZoC4wAd6nXMSDAMkLeaZwAxs2YmCPKXP0WwVhTRYzF2B/gg46g9rYtj2BU
rgBjK8J0ps/bdSvlrFDYpCRGqnYJGMG3GAJZRlStcTvMJ/zgku0UHQefi4br2t3Y39/UCSDnO1L81Mev4HYFQbdOnMuS9T
Fb1IJfDMjLBV7c79HkZkC1eLBJA5H0RAdb0GbtbC2lPC9NrL6e1KykXSQ8QmCvy1k811I4ANySa2YYYu5kkeCrwld6obZ
txnNxlgeboXPEa4Oi8k8OGD2HD+o3ZIgMtcmSytN6ZAsZzEM787Azd9AVanRnUXfWQ2NqN9BW6mrXmQ87c4/2Gj7q0WR6h
27v+CrXpZX26aMLutfw9GgS36E/3qRbaZ8+vqQB7jRM9ic5V61Umvkm30j79fGe+Ml7jRI+TO3Q7wK4rY6V9+uNF7F7jRJ
9G58L1Yp/epFtpnx5f0L2Gp8dJNO2Oe3QcCLeCd+I+P+kAvjYnlbb04+R8Nd6DHwK+Xxxq2g+9M7UStSQk5PsVwgMVVvuv
d2sjGSW1LS4Nt3O4KHOmAfRUOsAdyKU27wfa23Bcw+4ZIJXBcuymd0kYqo5gw6GEmspSqTTZ0c6n7T1n8BwAA//8DAFBL
AwQUAAYACAAAACEAlCPW3KgCAAA2BgAADQAAAHhsL3N0eWxlcy54bWykVN9r2zAQfh/sfxDK6IlbpY0wXXZJJmgKFrgzaw
V5lW3bE9MNIckk69r/vJDuOQx/a02Zfd/d9d9p4+vD1KgZ2ls1brrB06sJRkzluuqCqA6v69ecGHQmNJwa/eX4/eX1hwf+
DIMBuoAECG80GYt1OjC07g4h7OsqBarox2emsd/DVHg+y01U4Gr4sB9/nb0r1YuTs7fHj9o/7q7bHHzk5yOfaPPqi9U/3T7
+9ai9ZiAtBk7/osqUW1XiBdSEVH7NTel1H2rh6/Ul60+zg4C/oVXy5/XhEVHmF4i9jpdT
i9NsD7qHWFK/2Fg13Kd+jWASLvIASlvIASZZk/Dz7lPS24/+CCpdU6nTgMv/FlDqo9VT9dn9DXM1vX41+24dALbm2M37YXoQ
4MnJKi+rmdTwlpzEx8gUUKaz1ovCrZrVwRdo28n2sxnZTnvHjb5T6y2u14irraFFMia4IIU5TnH5QY+jmaSMRBLP3r+KlCtgpsu+ohkl6Po/pycW7dnl+QAH7
wjzNhxJTizG/RQ87U5NUKaz1ovCrZrVwRdo28n2sxnZTnvHjb5T6y2u14irraFFMia4IIU5TnH5Q49
PLMZe/NLvLFu5nmwPj5Z6H7MaoZOeurrSinb9JWZM5fRCqyUyYy9hng9gIYlddnWnVIfI++eG+AK+TQvJiUXGm0QmJyXqW
F6ZeJNCiLYLN7z+afwMAAP//AwBQSwMEFAAGAAgAAAAhAJCVG1V8JwAARooBACcAAAB4bC9waXZvdEVnY2hlL3hpbLBpdm90Q
2FjaGVWZWwpbml0aAW9uMS54bWzcXVtz23Dayfj9V5z+w5mHf1rJkx052LW85kpWWk1o69lpOtLZ99oGdoiZUZziyiH41j760
+DIMBuoAECGLg482R2YutVtz3Dayfj9Y5z=w5mHf1rJkx052LW85kpWk1o691pOtLZ99oGdoiZUZZziyiH41j760
+DIMBuoAECGGo48YutQV/Q+NBo3MFnf/uyWmafi3pbrqqc/2emDh7OsqObrVVdnM9+eX/1529n2bbbJq0W0+M7T+9vx+vz/+/fZpvy8bi7y+W1WWxqq7fnssU8M3R4koq+rmdMVU37t1t8iWEPUXr6gp/nssNk3V4qW6hDG8Vs3+qrJ6zi44OrD6+/VdVu8VW5zns+NAVfYF1RIeKyPMW6tdKXlypsl9VRYLJTMp9y4+g+wq7v3vN1tKfai5LCSiMf8/afVbdlbrP5ycPOJCscOQqxNDoWoOdD0ifJyQ7xCHqiRCpA0D0rtvHxX3OJCsY+yxvldrPM767S9Ov4rPj9IOJCsg0rbKFrZ94jJl8bGWadq+eirfawiBGqnYYSMzT4J/z5wzC1oaZG5fRCqyUyYy9hng9gIYlddnWnVIfI++eG+AK+TQvJiUXGm0QmJyXqW
F6ZeJNCiLYLN7z+afwMAAP//AwBQSwMEFAAGAAgAAAAhAJCVG1V8JwAARooBACcAAAB4bC9waXZvdEVnY2hlL3hpbLBpdm90Q
2FjaGVWZWwpbml0aAW9uMS54bWzcXVtz23Dayfj9V5z+w5mHf1rJkx052LW85kpWWk1o69lpOtLZ99oGdoiZUZZziyiH41j760
+DIMBuoAECGGo48YutQV/Q+NBo3MFnf/uyWmafi3pbrqqc/2emDh7OsqObrVVdnM9+eX/1529n2bbbJq0W0+M7T+9vx+vz/+/fZpvy8bi7y+W1WWxqq7fnssU8M3R4koq+rmdMVU37t1t8iWEPUXr6gp/nssNk3V4qW6hDG8Vs3+qrJ6zi44OrD6+/VdVu8VW5zns+NAVfYF1RIeKyPMW6tdKXlypsl9VRYLJTMp9y4+g+wq7v3vN1tKfai5LCSiMf8/afVbdlbrP5ycPOJCscOQqxNDoWoOdD0ifJyQ7xCHqiRCpA0D0rtvHxX3OJCsY+yxvldrPM767S9Ov4rPj9IOJCsg0rbKFrZ94jJl8bGWadq+eirfawiBGqnYYSMzT4J/z5wzC1oaZG5fRCqyUyYy9hng9gIYlddnWnVIfI++eG+AK+TQvJiUXGm0QmJyXqW
F6ZeJNCiLYLN7z+afwMAAP//AwBQSwMEFAAGAAgAAAAhAJCVG1V8JwAARooBACcAAAB4bC9waXZvdEVnY2hlL3hpbLBpdm90Q
2FjaGVWZWwpbml0aAW9uMS54bWzcXVtz23Dayfj9V5z+w5mHf1rJkx052LW85kpWWk1o69lpOtLZ99oGdoiZUZZziyiH41j760
+DIMBuoAECGGo48YutQV/Q+NBo3MFnf/uyWmafi3pbrqqc/2emDh7OsqObrVVdnM9+eX/1529n2bbbJq0W0+M7T+9vx+vz/+/fZpvy8bi7y+W1WWxqq7fnssU8M3R4koq+rmdMVU37t1t8iWEPUXr6gp/nssNk3V4qW6hDG8Vs3+qrJ6zi44OrD6+/VdVu8VW5zns+NAVfYF1RIeKyPMW6tdKXlypsl9VRYLJTMp9y4+g+wq7v3vN1tKfai5LCSiMf8/afVbdlbrP5ycPOJCscOQqxNDoWoOdD0ifJyQ7xCHqiRCpA0D0rtvHxX3OJCsY+yxvldrPM767S9Ov4rPj9IOJCsg0rbKFrZ94jJl8bGWadq+eirfawiBGqnYYSMzT4J/z5wzC1oaZG5fRCqyUyYy9hng9gIYlddnWnVIfI++eG+AK+TQvJiUXGm0QmJyXqW
WgvukY78qi+ViC+Q2y7NvlJo2PavyFWj58LrItzuA698PPlzsts16VdRZa+S/QXG+EVrPZ5Qyy6rd6mrViA698PPlzsts16VdRZa+S/QXG+EVrPZ5Qyy6rd6mrViA698PPlzsts16VdRZa+S/QXG+EVrPZ5Qyy6rd6mrViA698PP
fgtAnZ2ezbJl8bkAqx+dPX3ylNqi8lTqcJ4/FOubOt/c3olEYXF9RyyQSe6sod66jAVc29u8LhY/NSM4+5j/c3olEYXF9RyyQSe6sod66jAVc29u8LhY
3/wp3y1+euHFWwBLnS/euHFWwBLnS/euHFWwBLnS/euHFWwBLnS/euHFWwBLnS3/wp3y1+euHFWwBLnS

aIRMklFAvOqA2tO4BCoOnSzE4mYsyCvUPlEIeWWyoWrguxqz9vtqM4UmdW1p+dAbV6X8Ki+vojBiOkNgktr3wsbhqvJzM
Y7+Ih3/Nnn9pzTr9s85tCDQzFYZ82WaZ+OZ/9ue0LaRqTdCaGkJTrkZ302E76RiSdIFPUuRPd8/rH7agC9D4Abgc60WwI
Qb2xlhadbHj34xOLrcGRoyjZQyBA0f2DhIhKVccBhjNnsI4Fbcxw4d6ESMCoUxYFjlMmFpWRXanfeMB+1KcmtbhenHEDz
/jFKzcxUHRjAoNFKUmAURVxoA3KTgycXGWI3fxIaJZsRlGNNFDD8QBqbG9gr0RlMbiSPNStLs5bo/RMDHS/bYKB7VOTgO
zF44Dzyk0MlGs7JqERS1VRrdYlEgvKmOMNtPmCfQclJzkPko/zHr9gLFJjznvp6VAMFqUk4UVVxEE2KDslanRLC6NGKUm
oURVxqA3KxqI28niXbIBh3AghCTaiIQ61IdEpQBOHE6EUF/kSDnDl5ACXID1QhFnWwA5Wu5fmno6bymSAf1vU5XrBLQMP
8fOAtFLWcnhvaOLOmWFM9o9dXsP9MzGkFRWHvUiwamRMRtOr4nEzNXZeFozjoPykuMJFoGIbBKzJOQKypspoaAcVTIqt3
1EFdQQM20wim7ZsQqbQFFh10UMvIrdtWfzjRCZiHb5V/uBD959YhhxstCxvDBB63Rhu1MnrJb/my117V7hpr9E+NdeTB4
4Q6ELcWZe5BOnBZS4JTsAGOwilIbT1OflTcaLr6+F4QF/wGrY3bq3OQFioiPvhorTEYOOUmQCffxbFby54BG1vdNoMzEh
iLjciHwZwWJEJsGHDc+c6+wVnpCTSScdhofkhsrsrtPF9aLiOTY4adnaJA13Bwx5RcmbjfgFNqscZsaJ+zjbo8W3pr4vWE
Ok+g9GRoWsM0B5wm3954mgojAR0UnwxRNnZ1brBf7EJKtHaO3bJrPEIKmVk0A4JnPNEQk33L6ImFCe/UIwv7YuR6IKdE
GFiOkJYSyhAXpnD4FMsS3i06E5ePnasxGAevfAQ1jdiUUBFLZMMAxQq6ECJiN/Dag3R35fL249iGQ2jKZHigFgxAtlUnQ
z6oKIjqwSrG8PdLwarrwVTZMxqMHWn18OgpiOrCKvX5sFP7cCxNuz3bMdszdGipI8D2HZe6XRmQXX6bcwykoFMN5sdWE8
KEUpCEff946wwWaYyS02YZ681J6wInNS5mDTkn07BvUGNG3Qy5dELQB7X3G+Bypfp4Iw7RvgowBTDQrIigUvQOmO3quRs
7Z41UKyr80fXGlWIS7pkjwJIf/+DVq3GAzKp70GWWIODaYGUU1S3M6opbBOwPWt4Xqc6tvk6xJJgUtanLqGxJbJGy86oK
LM3+UfwRV6x5VpD7TtQzRFBbw2Qh7A3BcYD39Sciv6gniOC3x9vO3cSTOPBnBRvkSVWtRwDnHThzOnDhG0ESIk+C5ihaO
GXToIVSp52D/3SiMxkoc2JJ+YaAU6sLhpNr/ChwdzAWSTxNDiU4rJQP8iVbJTsuwXXSxmqurvZpzDScBVO5duyBD1XMHg
7+xKOApVV+8gNXBiu4VGguXgem00mXmOLmf7iG4Hb0q5MO1h+uc4u18tlewzMBQ+j1Fr33KM9Mep5oKxXhDk0WyYTsqgl
ikh75CiTMQXe/gCHsmPbuPonqbSXq81yfVeIA5Z9u7MP1egG3ZLMSnH7sUe9RFsxcAArGmObLcyjp1VYbv6mvsmr8r9tu
052ea3+wQd6XxEFAkJIQo5oADBewKEtlfMAbkOisajtfTVB2Q3FeAlfObgR0RQaensf23I7niMRQzazSDADdRwXqtalGB
vWlmUkXPEtGFXbg15qAutSMi2yXagkjVummdhFdFMKo7ZF6GAc1cadUrFwjRMIpTlZ35/gpSFVXET2DQYVO0IHScYGQ9s
yOwBMiZloJ/rRyryo6PFMR1MFq/jRUxMdnTK8bC4vBQbayXVRztbBeOVdpZjMLw0UptMtnDuk4M2ypWFR+hAfL1zW8v584
UUVW7f98gK3MLqAHFvJ8gNIl3DYWlDGuU6n8wWt+LGt4GR42N3EL0olJfqOlmYbhQcgnFgvSiJ7TmmWuvenEdIDQ2pqqj
oHeXmfK4DohQMbzBSgQU0oSVFQFU3CPQw3KXoI2ctN7CxcLruoCnqcUKyMIN0JIgo1oiENtSDQWtLHni9aoPH0M7hose3
zKJRKLypjzwWt4QrQu4YNlV8v8BvsRIST5EdEQ50dDorGIje1HYB+8TAt3uLbwRUgGOUpMRY9qiUZwUHxyFMv5b9mrAj5
gCc+972oyDr82aGkYGkoiIRyWnhzBBtb/rREYvF3QpaahpsUj8fLJxSI14iBMPOVAP0IhU1LBEcqihu+dAbZMLChjh7H3
ZbMkY/c2IQkXqcouoqc7dIlMjcqvuXxA1w5KlJKEE1UR50eDspMg1z/BCaft2a/ihT4EZGmS7qSS+Q0hRW2/niuy54R4Y
HpRa6Miedmg1wlojPDcLaePK6HZzkLlpkGmtw7NXHSib+lNM0E9479fwPaM+j3sKIrzmLwFP9KEUEHJ4QEHlw8pYNuGz3
OGZKfxHtGfFkGfJU6JPIb22MYWIH441H6q4EGVqmgyOWF5U8NHJGAlH76uvGw/e57XMPers/c1fPcdPrWsvr6A3A8+WcW
xhLtiogkSdk6YW+ji+ALK6c+EryZvVuP1I95soGtRT1y/fcNVmkUdr74s1V1Q4QxOqapY/cdcSy5M9G4Ty+Dr/jgB5Ogc
uZuswNqOGek57mBlx4w7jnavSvqtMJM2WsswFY/bLiK1/1FqRz4Ia6w7wsKV7Kkk9V5qqMvY3wNENNhhcR0H6J60lvJgAv
qpOWVqnBAkhvC686iXsmIXrKjDdspBXwSNKFI3XRxO1Dz7gn2QtkxB80R0zQoHxT4ji8iNJ7oO4mB1aVOCXwwZPCbyt1/
AtZ3HC+AKG3Tewp46bq0oLdwxGn6zLjsB5QogMX/VKMhiPwS+pKY2AxzLfkubu6E5KQaDXFwMALRGJw3azbcX/yagfCYr1
ck7cwL0RCEhatphgseIHpsIBLg9usWWev82r3CT55s6vJIitDTsGJUQNtVKyDdLXib0hB4hNiWMAWWw1FKecUPJSehBqS
j4LLKxeJ0xVcbajmZb4coeGpU1o4KKu0FICUbAw4bplIYPbd9enj0ascjjlnsOPerDEyKDkFHCQeg49XLBKiEVzmFRyNs
1tWn5qETK8zIgj1edpwRuIyZptCgTt7D/ensQeZNN94rndHU8ournXiJGNhGrEbv/1elEsrSsEfWqKK/XSUSj1llgOGI
nPCE1M1R6oUl+l65JMRCIOrSuIDz70fwX3906RycDJurmD3K4xcULUsCbWFRDJxbhPiPR0SP29IJMuZSwkm+7kPpzdOwA
ST4FIGHM8TSwzF/hU6UT6Pui0eq1yusM0zveY8JHh1N/WTJ64FmdK26UfRm1Yx3StT9om/IHM91WFI3IKblJvgqOhfG3E
D4/Wu2Ldbi++XZdwHx11fISQ0iKJAruobucaEIzEaMRh03X+qWjuslYjnID8XJB3rm1qCmq2lhjoQqQnxK/8LxmUX8PvJ
IyEnoimJ/Kx+aFFIXuXVzcWGjI1FZNOZyQyLqkJ8Wly8fWiBWxqbElMglMaPSEJJazAdgl3TCI524KRWI056wXLRjpM3P
WMUDqkM8aZekssqUiAxltRgqLseYyYwNKfIu6S/cu1Xe7HBAfsHTB9l9hSkMlhgyHcB1ql8zUjnc/ejWjMjr25oyeqr0V
CWoARmqIQYQUi28sIax/Xu49zZqcQJSfhgbTGoOITOzw2/yzKm1vSE8mUFEQ6XRFguCQicRijvazWTbmuwL8v4TMv4py0
uE01Fy9ZoZmDSYtCyZ2DRkyyOAJvlLwTwU4LxMIy6GWskOlnbZned0GYqbQ0rJR0H3jCMPLJTYPN6/yLeL+zaowrCTg9D
SOsIRanIdmJsCorHiuUnogV0hCN1YDsNFjBaE23abxmLRNN1OJW93XD7jOJjFVewYkBUyvz51K2LLWmmhB6V7RtxLSeWH
fT0Hk1TI3he/h0I+4ie6sFZV/sWu2pLpe5pCfGzFy01T7jfeBHowpOpP9Wtz5RAuC9XP5SlfAMw8+5+Axyzz0oOQ0qaEK
OBg19apoH9fKxrc4veTiM3hXbYgknjoT9i0Vd0IN9XVI4OIw62awUgRt0KhqywRbiIdGi4w00tUqwoap2+TJ7BwvV1Y6E
H0pJw4cqt0tsNjG3YbZsLFryCDc8aZQnY4T4Rt81qlOGh3qLXp+2AmFdulHkbMJTktXt/n1W/WYRKdmISUlu4Ck6qdgfbnE
4vFqLt1N5JPfb/bio3IrextUQwnhDSsiOootyKZM1DHYjZG67sqazjnJtE33+AyaU14mUqYcntaYoB0LGrjetqr3AmfQU
pCz9ARB96w8LTYvYa5KzyGJZ5mz16sxPoWjv82NQlBW00ciEHyx4OjOfXB5sdNflQnAHCZStIRRLMfpV90MtPiJ6+hiQ+
s4ncC9boFQ07yREZPHJBhCmKxHKMXkTGVfWrPICUhZ+iIQ21YOBYxqXGs8a/U1n5+BIc/lLwHZlJt1CgFZcwAPS1W8nFA
NJ5rE5Lwwe8M41DkGYy4RGIxGaPFwfqKuHA40sFOBQBUOFEc5TgDklOgpMuFXEalmV4TsQqtVABc6M8XcKtM/RyYOSm2E
RrYGM6kzMnEjA4HIULwHVVQjAgRqSzKg5QWl2isC40bqLV1Ui3b27l4YrEzs0iDcVjLUSFq9YEGmi09DcmEfpDLfIT2ek
8+aY77tfVy0O/eH1KMuOU6RumerlFpwdsbKi1lgD9GVBPLGdmbTVEVECxzTULLZBQShhOQZd/Bg5BeMbYl7IzThc1fKMyJ
NoEEQC/u6W8yHHu5xvNl1VCkWHvZjzK2zErPbcU61DMFmvwqFhj32Y1jx3s+Wj31na9yqsgvm1+9vS2wpDlZLXCUKz9LL
DSxDSK/HlqsfoWKy+7krLnftykPKjhn3djOnW+hub60hyE3aaC3DVDwQXVydB646VBmR2v8otSOXuXCTwAWV1HupoS5jf
2zZo46C9E9aSxLoi9u8quC2FdxMgPdFtpn6bbWYjmBWRsRMmmZgzihrgvBwPT1hsKx2KOFxpqrG6xSk3ndFDt9C0BZ2Py
1YZfreqPK5YTgkhxvSGA0+PDs9B4HT2vLF0LdEE1j3qB+LGrWGJzqYLRJMn5pjQZSMTkhZfaMSzGhgp86IYZZI3FwqJsD
serda5fVdN7Q47LvBEsFYC3AI6JYjuteOoWKyR/XGz8FXRXxJxo4bfC7sPAbF5n0mUAMouuY3UmzPKopTfry1g8EYbwDv
rxkrz/GaTpTqP0atjDtwD68ZMq7GuKpgHBvWsI4A5X+I2mF68TaweI94B9eBe0EHQcX0eL4v4EF/8WVUPXLpU1CXIMumS
cPDIc3K6MWhRfHxH+kwso3S5cNW5Srunoy1ZjloK7w6AAe/6Hu5hpBiiR249+VRE0udolQCdi9g25ZmmIh6gM7jQ/+Her
3b2GstPVAtw5jIyxzHxH1I4/Gh/q64ERd73aEEDhIIDhP36JUVXY92kOiyGLUFdOVyV+7BqwJuS8OnO9ei+OpPjHqXtg/
OSi3KYfAVcmTLaIH2ttzIjxyKN8iXhXi5hTiYoj9Q1FnWwLOd57NTb7+lxABBJdi5DNxFL9cLduIeIuRwBS1q9UEqe6g/
0WWczx7Osl31Eb7NWs2Lhfh58vzZPJ/fFj/C93ryen57B3R07exa6+7LIq701g08ogynP8UGrR8yk9v0m0RAThVJAA8qO
RLAr4tVsQ1H3GQfC3JTbxrmg1qOBPQA1xYsY4HbZid7gbgoIdugKTktiHDQl1zn02HkgaA4IYvpq1FganMzAbBW5YYEoh
HDQXVVrD4W9a/5cleIQ84N3Jk+nz3dI9jCM/rWjTWMoqQ6kfQs0BMcukziwHMI7QVgcq8kgIKOyDoIR8FSHCMAplRFRT9
ipYn2pMD9syjI+VULN8EwDmxtVmbh/e20rVtWblLQ3H1DV9Ej9AxIU4qrsf3JNKBdwVNMOX2JVAWhB5IWPa7uVMY4k0Mk

GhJl8QgjaqnKGoXi2ZYBlMm7Z8N0GBAPq2mX5bFHALM18PThbDKPA7Sp1YLJFw27yhrUcQRQu+NjV4gR4iPSlALjOPFRG
oFHgskDmXYE459TE5Y9PZLoigLQLxntfaOAJ4YIfuwwx57QYVVRyHkFDwjcegdPImbqA4xQhGsjBUdGSjKh80w0qCCTi+
xmaN6OjYQ9dDlwNTRaS2fpbmlottCFTY8FnR1TiZZu4hwzNabqbOClASOiP6Tw0FXQLQio1U81OMELlYJFL+sqBhP0Zni
x15XV0BAqVI6HrpW2XFaV1+262W25WBSVWMAeWPg17VOjDjeGimMEEJWqweDqtNKEf2IYu7FKt3BuYUjJ6QBSPaHoDUjF
QCdVjeh/nW3aIVDHJOq=mz5mir43dEpRJHZOsUnBkwM18YCg5XE9KR2yXkcoWh6JGKBG8a9iWX4u4AyqGUK4Nop5R+k0s
EIATwVu1Q8N4xmrwAEvUXO//QnJChVZNx3ctDFzD7hiTXFZrBEjrnSmQ+7UcHSYG4HeCTjlGwFtqjAa6gHxJJylzvuIJJ
212i2GYFaM4+GsNKYC7ZQ/MqRRd+IEuecZAd9eWTS0HtEkVMfx3A0c5BBvyUFxLgv9g4CpUk34IiakSDXJRw7Seyp3Lqa
nUhs5WReQqggwCgo+OhQ+U1IPoIF91+3dP32ALPs7/bwmQzZRdS+nDGQj8VBMHJKK5rDTVsDDqdVYYL6pb/Kq/G8uPsAx
hnv+UKxv6nxzK04U679NTAnBRDPCR7UeKzMJjKLz61OKapvKSfPA9josZMdA86cKDhBVReO4GM2R8ayBo+8BN6cOXdPmy
FAx4twxR+LcnePbLwe+2rz53EtN4nYGoJCfKHTj9D1qCqsxc5M1gjm4msD0MA080kTPvSCrXnIAK3VrjAo47vDtVC0xVG
QOP0WzrLJFeeC0gnsFzRFYVOaYjAMLR9/DXT11qNlzZMBVBBaO5KsOXKD9cvDXGpvPvdSkHJ+4rnpKansBCtcfSt6j2sK
zlpWFctXYQ/ZRN3GRjs4FUArSylePz+L7qR11FASMvehOq+sFIxT3+wNcJpdZP83w9oZWxmUqqyLytKpaXzOFHSjrQluY
qtKNMThii6mnwX5wFdtY6Cp9nU9GwuuUPhp8jdUcFlzKsyeyVFkUrAOi0ZhKfWN7bGelrnofpIppHEyVthRQnbJHgipas
WEB7el7YtkrioLRIxaN4Bge+Rpe6tnVhbjWbMyJ7A8VsAyzbCVVyPtV5guAHbG9uXg+oxqQ/QP72pyV8rUI6JisjzrqXC
jLZJa+/NLALbViwXz7QZtq8Exm63v4UKan5nvyZBbCc8Tiq7lweoV+xF1DiRkms/L9uoFvZKmXwp222myTWQzfXhNXKRf
DRrOck9ktG/mF+VUV7Q6InmZjqwBej9BHx8ODlX5KxzKPUtIMUzqwbdm6Kq7KYgnXRE9nz599En9uf9nmN4WyXyfL1C/d
fdITxPr82Qm9Xuq/boqiszEdtyMJyxBVfKohqVq0CtpB4Amyg2UyS43+gTXV4JnM1r6TYM3syZNZiLsI1kbMcGgrL9U30
O3WY5AObZluFHa3xaJos01mMdtnsUaznJPZjXov1lpET7Nxj97NHT0pJcowskKXFN2NMIhhM0gHNoz1LGwey3BgI+02iy
20qQc2T7qHHRlx+oFN6vszqpSeSB3YHNx1YYNw+oFNQkEKW4SSEwxqpfHQN3xYjp0FFuNhqobNsqkJ1r3P65uiSTPP9m6
ZEmXGVVnBczFl+6HmzbpuyuomKZq+gPO2YvbwzngcAqdHGfbyyxye3e40pgH0slqIa2PirrVplkE6tGV61gkLXvBVXpj/
fyrJxz5ZhigrtYbWT5PqVKvAVmavwWVL8laXj28ym5Xj/VKVDQBczsk5eW2yzTa5xTiw4IBj2YwZJ7MaWla5zpp1JrbZu
4UssbOuv9+hzWY5p7f7bV7DIUllut9yynto26EZ/t7cZvCVYLWkAx+G2xK/drAc2lI9A8SRA0PLMkRZqTWkkzetAlvJxD
cf32Q224GLxddmm9xiHLa8NmPGyaxmoxZrNss5vd00Zvktp7yHtl0vW8GxCHhFsoHxJDaXIR/aQrs1YQNtapR90t27E1p
JAzZlAG44nIGYfmATcajFpuH0CU1ioj+2rCMf2MCXX+CjjdtCDLLeFZ+LatdOs+ARVzTKcvEc2FQ2BmI7WYapjKThjjWT
shzYUD10HpjE9OQoA7X6Nh6giBN+Pwbt6Wlt7+u82uZzcVMEdt/hZVoMrJtrQtNlPJRfZbJNtqlRpmJxvI6RBLIeivZVj
tFlyFG2avlRPKK3MTP7bmy0h21C4zUUXnd2c01oeg8oD30UadJ97VFJkvt6sbSIB7ZTvKWwgNcmSLzSibNsW8Ab2/A4+K
ZdMAC+89m331knl3EYpc+Foce5YUyx2wS80P1z8btamMhew8NCy2129a/s7OHDM1yzbi7W6O++tYyGwpDQdH+GPwoy/NH
xGf44yPDHkxn+al3dZK+KXB/MIotwNnUyQy/WdaGdum0IGFmbyhp6+hAeu1/QU6SjO/ErsTmjrw5RPCmJt/GRHR1Gt/FH
OOSIetd6vVqLgRcx1sXDW312AGTVNaS3f8pXm7++wg5gkFgbITzcd92jge3PcK/2p2q+XhVZ+wY4tran0c5qJQ87d3F+r
E22PrfshzV9dNcgHdgaOJHQ7IjPIXSkkVpUztdV24OeQod672C9hxX+BV9tWUvTNrQdvDYO/Ox+bHuzgd1SsdcqAiFs/d
mOZXJoS1oLD2SV6WSmUa0XTmKZ7XCWbV63g9f7D4Sh5XuWoRN7YLdytPX5oOLRqB3EC1WuVrDrISQsE1nn80Vt39TeqA3
x+CPloWjqmAgj3PtpOX0nKVdc5TkD2yt5Pm3VPXomn7PlnR62Ca20vdRh5zSe6jDG8lYfH0VXe+y99eLvimZXV3CVInux
3cIM2vZVk4NaeD/tyMzT8k+WYRLLbJ+0bPN64ylMVQ4EouWIlqW+bvyb+7JT3RyTIbMbTHaR0xy9DfFqG5kQ6loFDD9M6
cvd9oNhbmqtbuy2S9yv6ZZf+Cz3uYgVpfa3u4+Vv8I3H8VXAe0IZfNQYO+nedm5WlHKwTKRdbaHMvb5Y5W2fORFAcYQyy
ldPNqmtsnrRnRvY7xuhGntZFu172WkVlMP3b/VOLO2fWCAldqp0YVVwYPZbHmC02RfbLL8YX+U1UEGs5PC6Rqne+mQVE5
2tVIKtULX4n10NCpjq9YIwWGQNcbYv5Iu4AR+VYjXs+SpExsqnsNhoQ2Z3j5tbUVbZ+HjCtMCCzuWwWGhheEIBuoTDwpF
0+VZBmogjRdaIB01rUIZZVetg4Ua5mkOWn5EK63K1ZmocvjCmFW7WjrdROjMuTPZVofm46OQjl7XrqztKvdzUjP7mrdW8
O8RVssBXCaP5gf0iQN4Ifb5s9825VY/k3AGzydAAvmydr/5gU7o4URj96ofnsHJ1+2mgLNTn4v/e52XcAYWTvu/EXer4H
aFnNdtdT1A2cUpgl76XaGvcMlDCuezWfZZFLf2fPbhdf+ET2+KnhDcwH6Sk6mLWNt2l8fJpf2pEdstTjZVhb+3770AhRD
rTQD7BV6YTsH+CILRJh0OzB43FlbTNA+4FqsHYovXB7TFHAd3NzwEH9P3d3qWDPfFwvPe5K09g/LlnUiYGQa+Zg8DX3GHw
a/aQCuibMLRpuBAoWzqqBBfDkVUEb6anMhwCngpxSPgqxSESUjHmKh+qEpsUVBmiw1ID6v0juWmEB2qL1QOyxeuD12IOA
dazTIYw9nIFwR3aibYjwEwfc/R2n7KNynjUXxi4sXpQj/VqfLhlulSfmK8efHIhVdI3E7VciGqCIwZVwLj+bpvh8XiG2e
PzDLcPbYY9BOSuW7BWwBDUbhYDcDnOFu/HoVEjnFSqYaRXZF2nuK6A4eWnT+W8LKr5nRlyYvzeZRicrLGd38Xs83ynjK8
inEIh1dFNGxH6OsUAu69vjLbPu2Oglbz6RQcbTjW/9TiwYvGBpXhCsDEWThBGFmVvrMx1Fm/8pcz9O5Q2ahhW/YLHlom5
RokyH4YmbwiWekqs3UuPr23S3mhqlW351Yl1x2SQMvvQ1JzKizzOaPH6ELWYQYyAVTY+5DY/c1MlhAKxv0+OWDWEFHrqkN
+3N4KntDsHZYQ0bPB28vtjpEvHh7pLh4D8Rqx7Pn+nXh/UKyJlYAdHJWQW3KM5nL+btM5gzsizyoUvFJzEVIywCmDpUtR
hKVDLWotLEWoKp5hJepDFUiCQs3rJwovjDgqYOTCPKMIE1SH+My1IJYU9+sgt5cf+RLF7ZFs6Rtp9l0t/8ttTaLNRgm85
lpT4OZahXyVilSuPU6I+vGHp0OlakEzlNtPll8p2oywLWzJawgLGryv/sCnF/B5aifKw4Px8fNaFbpm23pqTPqwU2I2uV
jLNRaVSlVIO/hGOowiSsDqdzKruZgaGtS8WKFCPTFoAkrxbYWmS6oadjZjSp7ttQpJKxHpXGFUrRuE+oOFRzrFx2HB9ng
hxedOMBI09Cw5kQglurvI3KKpUkW2cnwkAu81RXK3wNxcNpZ8grdOavv53HlkpT7Xw0iYULnoqA4y1rU2mXTLQpVs5E9T
ESU49KJ4pUImvQegfPV7ii8TWlEq2UJHSf6N4M+t4u5ECvvdvo7hcuUFBK1wNb76flG3Ex6HxmECATrLeT7mc7aE9BD1k
5Kq9Hx1PVQ+OVZYY4oEU/5qcKo1V0lAH51rtR36rFJYGX1oHGyt2kDMibuWtxX+4tTfZtGDuSzOcrWWRQsEVVuk9WL4fp
mpc6VbpPtn/ZUNWVlFXpIHuCHU/4d9679bdPhFvnG+OoTHs3DT5qum1d+Uw0kSGmb3gm2LXUbau7iWtrokyO7CjT45Dsg
myC4xJs8Wh+p9/xXGekfKchTI7yUU2PQjR9G8IEkYstH83vNMiq0ydBuoLsgguQAXadOax/FII7ZXKUkDI5cKdMjvJRJh
fuBleQVS7cDV1Bdrlwp7pcuMMZNdSeHZVDmRwlpEwO3CmToz1TJgcIlMlVOQZXkOmuyjF0Bdnlqhyqy1U5cPIGVY7DeMr
kwJ0yOVo9ZXrKN2fK5MLd4Aoy/TTI91NHT2Hk6KgdyuWqHYPL0SqehNQOZXKUkDI5WgVlclQhZXJUIWVyVaHB5ahCgyuo
gK4qNHQ5qpByuarQ4HJUIdxnHW5glMlRQsrkwtTgcmBqcDn8weBy1LXB5cCUcrkwNbgcmLbPpOixpiNDyuSKuO1DEFqVA
y3K5KgeyuRAlDK56tDgCrLqNCxHRx0aOTogpVyuOjS4HHUojhwPNwyDyxGSDK7T1vwTMWmBqcym/LxuLvL5bXFZfCrhzW
mYoDz/fwEAAAD//wMAUEsDBBQABgAIAAAAIQAIKrfqzAEAAANgCAAASAAAAeGwvY29ubVVjGlvbnMueGlsbJJNbxpBDIb
vlfofrLmXXRK1agNDhKCRkLICdUnTA5A5dhxixT5mPlmaXh38dL2yRFva09tt/Xj3d8++++QdHJGSjUGK4aAUgEFHY0MjcxP6
7sNnASmrYJSLAaeU4YRK3k3k+fvxqjqGgGDpzWwWeKeZ2M1UYK3DK3g2/fxqjqGMGuhhy2WKeno+UFFsfdDLLLpJXmuNUqitQSKpp+2iNm74qosPxVe2SAmb
8aBNWxEQDT6zjqWnN1sHhI73d3PyM+6CVbfixfzgrqDuPIdNdNdNy6gyq7GNfyq+D0UO7SYQgdNNyNC7Dgv6r68hNdtDNY7R5ySklC4uwi3JNHKXKP7T3AjHt52MiLDzOz6A1H
+YJxVM1/wCz6mwWta6XGo5qtcN8gmV7vr680hdVVrVrQYV+iwRVCi/EkUSvI73/Af9h+uf1/UZ71AUk3F0qmUCCYOj
1r19+ezUPw1J+vcjCFnzpRl2RcXr6D+CdLkGQAA//8DAFBLAwUAAYACAAAACEALmnZZSkBAAA0BAAAJwAAAHhsL3ByaW5

50ZXJTZXR0aW5ncy9wcmludGVyU2V0dGluZ3MzLmppbnuxSzUrDQBD+0oioFMwjiG+g+ALaJJKSmLBJaK6xGctCuhuSVNB
H9OjZJ/ABvNbZUIMnEXrUWXa/2dn5YWY/7CnWAQ7fUNj2FrB4vZ/oo4rxFMVkwmhOIASh50Vo96xnwq1dDoMmu8Ety848
ghvc5edwrNCe4uX148fSx2PUV/bR8K/8kQl8/3mHL2mUzU3rDlMsgsSS2avR8X5gLp+hQIKU0WXLEhusmd9qeFmwpxz5H
qhm099IBT8WURrnYuZBeKkbhsiVbKkzWqB6amsqH6VaIfZ9BOtyRdlTQ5gn3m1EFeJWkurLXmqFJBaZuA4yzHRdlz0NIY
I6XW+G97gxcIGkbKhN5TMh9LLME8Zft5GuCJdX901j+vuNTNkpKtyFxXP5BAAA//8DAFBLAwQUAAYACAAAACEALmnZ2Sk
BAAA0BAAAJwAAAHhsL3ByaW50ZXJTZXR0aW5ncy9wcmludGVyU2V0dGluZ3MyLmjpbnuxSzUrDQBD+0oioFMwjiG+g+ALa
JJKSmLBJaK6xGctCuhuSVNBH9OjZJ/ABvNbZUIMnEXrUWXa/2dn5YWY/7CnWAQ7fUNj2FrB4vZ/oo4rxFMVkwmhOIASh5
0Vo96xnwq1dDoMmu8Ety848ghvc5edwrNCe4uX148fSx2PUV/bR8K/8kQl8/3mHL2mUzU3rDlMsgsSS2avR8X5gLp+hQI
KU0WXLEhusmd9qeFmwpxz5Hqhm099IBT8WURrnYuZBeKkbhsiVbKkzWqB6amsqH6VaIfZ9BOtyRdlTQ5gn3m1EFeJWkur
LXmqFJBaZuA4yzHRdlz0NIYI6XW+G97gxcIGkbKhN5TMh9LLME8Zft5GuCJdX901j+vuNT
NkpKtyFxXP5BAAA//8DAFBLAwQUAAYACAAAACEAvYRiI5AAAADbAAAAEwAoAGN1c3RvVhtbC9pdGVtMy54bWWgoiQAKK
AgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAbI47DsIwEAWvgtKTLejQ4jSBClHlAsY4iqWs1/IuH98eB0G
BlHqeZh52JLx1HNVHHUryncETZxo8pdmql82L5iiHZlJNewBxkycrLQWXWXjU1UljGBTDb7xCEqPHbwtWm1wVhd0hjsg1Rf
MT27O9XUOVyzzWVJIfwgHm9B1ycfghf/XMcLQPg7bt4AAAD//wMAUEsDBBQABgAIAAAAIQAMxBqSwwAAACgBAAAeAAAY
3VzdG9tWG1sL19yZWxzL2l0ZW00LnhtbC5yZWxzhM/BasMwDAbge2HvYHRfnJQxSonTyyjkNkYHuxpHccxiy1jqqWN9+pq
cWBj1KQt8v9v9YYffuKofLBwoGeiaFhQmR1NI3sDn6fi8A8Vi02RXWSmjggygxH4WnTf8
Y6mSmEq3Usnidrfu2HvW2bV91uTVguDPVOBko49SBOllyTX5s0zwhH2/kzhGT/BOh3ZmF41dc3wtlrrItHsVAEIzX1kttT
7wU99Pruv+EPAAD//wMAUEsDBBQABgAIAAAAIQBcliciwwAAACgBAAAeAAAY3VzdG9tWG1sL19yZWxzL2l0ZW0yLnhtbC5yZWxzhM/BasMwDAbge6HvYHRfnPYwSonTyyjkNkYYLvRpHSUxy1hyhbad9+pqcWBjtKQt8v8v9YYYe.AODDQqjo97H0cD59PWxxA8ViY29njmijggygxH4WnTf8Y6mSmEq3Usnidrfu2HvW2bV91uTVguDPVOBko49SBOllyTX5s0zwhH2/kzhGT/BOh3ZmF41dc3wtlrrItHsVAEIzX1kttT7wU99Pruv+EPAAD//wMAUEsDBBQABgAI
AAAAIQBcliciwwAAACgBAAAeAAAY3VzdG9tWG1sL19yZWxzL2l0ZW0yLnhtbC5yZWxzhM/BasMwDAbge6HvYWS
onTSxnkNkYLvRpHSUxy1hKad9+pqcWBjtKQt8v8vYYd7mNUNM3uKBjZVDDQqjo97H0cD59PWxxA8ViY29njmijggygxH4WnTf8Y6mSmEq3Usnidrfu2HvW2bV91uTVguDPVOBko49SBOllyTX5s0zwhH2/kzhGT/BOh3ZmF41dc3wtlrrItHsVAEIzX1kttT7wU99Pruv+EPAAD//wMAUEsDBBQABgAIAAAAIQB0Pzl6wgAAACgBAAAeAAAY3Vz
dG9tWG1sL19yZWxzL2l0ZW0xLnhtbC5yZWxzhM/BigIxDAbge2IxDAbgu+A7lNydzngQkel4WRa8ibjgtXQyM8VpU5oo+vYYWTyss7
DEJ+f6k3T//Cr06Y6Oeh9HAz/n79WGUFIuNxVFO00FQLIKIyOeh9HAz/+bUD3YapDBFyaf+gbU+ZlR82DYN3+EXuFyDKHxHa3yVgoXXMIP7p
r9cd/3QsAAP//AwBQSwMEFAAGAAgAAAAhAAZP5d9LAQAAgQIAABEACAFkb2NQcm9wcy9jb3JlLnhtbCCiBAEooAABAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJSSX0/DIBTF3038Dg3vLW2X6SRtl6jZk0tM
nJnxjcDdRlb+BNCu317abt1mfGe8/hx7lAMT/IVovoeG64RWJcqgSfEWgmOZCbuUv01rEMQ5TxWntZQ5TxWntVZQohYmcle3NwUzh
GkLr1YbsfSAiwJK40SnY9KxXa0dX4Y32u22iXLrqmJPbjPNpI8Rmf4xny18aC9XXdmiPRbjNNbjNPbjPNbjNPbjAASigAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAnJNPa+MwEMXv
O6NNLSUJcgqJd3Swy4bSNNo9q/I4FpLlo5map9+xzJnP1zWZ1G8G8x6Px64+i
IyJBNK42OAhwAxZ3+/EmtU2whkQPMCJgIWqidkl2hoagzOWAytVTI0h3h3h3hqadjFXlLDxE+95AILnI81sJe4JQQnnVngL
FmLjs6H9Dy2h7PnzZHloG1uq+bb2zhviU+ruzKWKsKPu6t+CVnIqK6TZg35Oj86VnG7VxhoPKw7lfEISp4b6glMP7S1
cQm16mjZgaWYMnQfPLaFyF4NQo9TiM4kZwIxvxm8bbN4QhZAPQmAoEBgAAAAhAA6NRV4jAQAABwIAABMACAFM
ACAFkb2NQcm9wcy9jdXN0b20ueG1sIKIEASigAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAApJE9b4MwEIb3
dwEYMdHIb3Sv0Plnfi4zMQVEwQaq6dEi7IzAJEraR7Xygqv+9RmmqdOjSbnc663mfO6frCx/QiSndS5Fh
dwEYMdHIthf7DL/uKifGSJtatPUgBcvwxDRe548P6YuSlI1OmZxphZhNAZPhgzrgjRzYHxWi/sWNhJJxWvjW3W3Vnsiu6xtWy
ubImTDEA4hIc9RGcmf8xuErb3Uyf0W2spnt9NtuGq1unn7BJ9Rx07cZfi9D9vLMprNUrUWezGoYz9qoPCX39S3qH+LZRKYYt5k2efubCBaw4QJHQII4o+MvCLf0giLezflQFNIz
oclP+5kfmG11/MP8EAAD//wMAUEsDBBQABgAIAAAAIQBeV9GgVQEAAGUGAAATACAY3VzdG9tWG1sL2l0ZW00LnhtbCCi
JAAooCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADslU1PAjEQhv/KhjNud5LJMlBqISAxohrtlcBv7s
faDiL/eLmQhHEi8cJL00JlOO337TJvCUDlzxY32qP1iU+OQgqtxKRLcGdWPu
VGEVcxi7UR2hNNcx2SuQ6HNQmHGFNKx4UHFUDIZ4+o+o8TJvLJBtDOYbzStPhhXhhNW98EB6B6R12iA0UmmeZVkODvDHL8
Cao6zoEq021hcdzwI986hKuWGTvfaX1YrwwTFttU2ZZh/YqOwm7/H8UUqR99Osad1kFKQPfguNwVsmu8lrKKXgz7hZmE/U

xW2OdyXHPI99v3fHgex3hJFkzv1l43QSq2E1k42oyfKJ6aXEyGmXAMbMMwpk1z0IGedGd28cmB+ss9HvnaYfC3OhH1/QG
e9+/zT9WJgL/fPSvzlN//qf0ydHXwj9BQAA//8DAFBLAwQUAAYACAAAACEAgGM8eBMJAAA5KQAAEwAoAGN1c3RvbVhtbC
9pdGVMS54bWwgoiQAKKAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA7Fpbb9tGFn4v0P9AcJ8l6mZdjCi
FLTVYo04b1GrRt2A4M7S4ITkMZxjLKPa/75kbr6Is0Q76sgkQR9ScM+f6nQv97qdDHDnfaMZDlqzd8XDkOjTBjITJ49rN
RTBYuj+9f4fFNWaJoInYPaf0Ae9pjBx4+Hntuk6Mip+VQ7+imK7dLcN5DGTqVOXbu+3aHR1GY/g7ul1tZsv5ZjRd3I630
9ls+fPmw+pm/mG2uZpvFjbJu2fVlqqQFe6ucN1SjrMwFUqXTUaRoA5yEvrkECPHsEnygFkKcqrHxqxSNn+0WKLZZDZe+t
PV0sc+WfiL8XQlX86u8HQCN4PdEn6NxdrdC5Feex5XVuHDOMQZ4ywQQ8xijwVBiKk3k3mXJ8kwC0kl2z9ZSjPozSDKT
PREi5Yn4jRBb6uaDcff/jD+8OnFxrqRyBskcqpE94ijAofLnQ5V3KWBljoLvIcqo+BiGNCJemmxC0XE79OaWL8RjN6GhF
8HxOCEKr6Rwt4HjCJzpitA1BzEKep6en4dN0yLJHabKx99fHex1t1k4Hfv7ZtDh6gW+qamr5QNy1ixc4WKFgNliMFuPB7
GoVDHx/gger1TwYX/lTTEd+YfQwTlkmnKQ091nknnUajahMGsVg7VYksgcgstKIHmQy2kecfs0hc4vPdR42/j+iBD0q5o
WwR3ihKLJsLZuMBmtXOY5E+DOk9R0Bv4fJbxjnGfhk5LakP0LyRxb1oPokE56LC27cocMv9PmJZQT+9+/dBxmcZ18MJBs
k8P4mOl/YCs098ulRQk/mozGt+n/D8uqZsbc0v/pcdev5RAqKrAu7IOCsmHSMQB9YFm9pgPIIsv5rjqIQMh5s+p1TmMRl
vr+cx0Oo9wSkCLdipvg8RAiTgKVI7CUGLbxPKBMJzTZQADMWldDazvKzQbVT0CKJ+jE/LXjB3Ca1BKi1+7ma0eg6TAg9r
N0l4HQYRciP4EgB8yTkaYSe68Xduds6f6LIVAJSLcK7PXW+ya8cFjgCPljNJQ3iPHxMKHEEg+9C7oSCxrpC70NCKDQkxc
UhdB9YzgiJ9c0NkqPTktyR6NueLwAcjRrQKkhnlUCGxbBAcH3HQDALsekcPQsOXTrkaWTmrZRJ32U/DW2HC1UUmVGW0Zjz
AvRglshBEYfKlsJI14YWWssJ1GKsBRlJN80gGPghhrQrGoolsFysW/OP3+6KeVBGrbSwlhUxKbXoIm5C0q0Iz9Doit9Lv
uY5mKm+UPk4eW8ZXTI6DaqFTgbtt3c9B37a+ZaRWq1gRImPoJF9KM0MIGdMOkV8oTWUqgTcQIRdGRF2ijrjgtWyQCh5Po
lvGioqSV+RRR9EubTU+aiuBDixhsU1+ORLUwaKTr6VU7YEGlvKstnUV7nTvRBzTWUBVgatU33sHDezfk2kwWUxG44FP6X
QwW4zpwJ/QxcCfzwKy9BcYTZf/VTT23o9QTMMKxnGeSk4j82cgfzb+sd9pPoAPDxT6ogtoUIL3LLuIBEaMGhJzY4AG2h4
BkC4oUJ1higF4s1iasdpGglEPtqkcu+91F/Tarqidl+DoSpNXxtjkaIzVyh6QqoBzFAMH2kRnw6I8TpRTWpUrgtYVwmNB
J2hKKR6Mr+bzwQzN8AAtrsggGC0XfkDGMCeZ8GhFb01QvmdPKmDXrn2+hZlS3fl1E/Qsnl5SNOayOfst4R1Ffxu9kzGHDvG
fuSp2fWANMrlFXAUFddnyc+y6ETabbtTbT8R0C9Ejy22y8jaPqKCAKEA0z5jiFUituAe3lvR2iVNP+PLzNg66bQpsQrmw
bdkpw1sjUGDInMMIAFMJshwdXWBLY7X2DCby2cMjqo7BFemuqQXSTVxr48qQx+fsTw12Ii/BfMtWbKao1Y/aXsNVxCRTV
m6lwmDcKEC6iEsIvSV5ByMEzzLFKWJNgzZYd74+HYK8/CJFIZS6sE6pviJINOvRDm5PRqIV+67cqzn7ZmaYnkvFJga+dk
WBWkIbm+/J5hxaqQieQ+TBaYZVRpbET0wCbc+wr6gzRTbzTzRhOP4CGUgXLJc5YU1hxvcb3iVZehPoZJNT4V+1A7D2x2n
xtfFLNEpfaZTVv7sB16zNqm2keoJobgawxIKlh2so0p7rQiG2K110bQrnS1RqiLw3UIc7KQy5gLJdCzf20TfaIPOzJdqT
5brQr0Hr7WI5mZ0ZHDt6ppjXLTqY4IRSQpultCWanMjmKmWNcu3in6TvY89/9DsTh5QSdxZTi/jIG29Re9kTzd8LYN3Sl
OhJLHHLC+jywABPSRZc8nac+QRWtmXpy8DbOMfgvleuFCbkWmJgkTCubsE7svsQ+djj87uYLSyyqIr1CaiKudVZLHPs3k
Boujb/CMZY4Vkg8dueJCaRpJArkVASYwIacMdiSwO3OgODt5Cq99YCWht1z6ChRA8jsU4X3BbPjjD8dE0yNQUwv9FNW01
W2Jfnlgtbeioz0VIS4+wts9wAty+2rv11AsNmxP+q0VzKVYn2BpI15AwUpEgkXpLoxpWZLqmhpke4BYyHulnDLmS+t6Xd
m0rV/1ru305vYN+jXd6LznBrzY+hedRGerZdqWxspUbp6K1sbuGWWtsJasvcwqiO3eYFFuuZRrRpvYF9rxhZuPCUt9gDKN
l+3XSBbSmrFWLVIsa4ue6ugCRnysthxROPjLSNWTVaVHTU4/Lkqleb7YuLJmZdqaiaG9irenl5LfbzQ3nDIeQpuRn6FDE
sw2FhhGkRqfdDbwMh+6+oNwXtOzSdARcWMw3gL3wXhNWUfqCX+0bW714Ks8Z95yiO5vk4ZnDe447M4LIK88mtSaFOtNFd
1aYlXroQGtqX+BsPdgKspY5mmxs+90M15c4HLFNT1mapjqTjQULrVAZeUbji2NXW7aWyLXethWtlSyux6YNmN9pQDP5pr
8H9Glp7ojcM50Er24pCGxIe9PCbqw3rWz2+8p89PRJtEyPOyqJ6rBzx9OWwaRxksKHi6t6spK/7E4Oz+xODt/sTg7v7v7E4O
/LyfeqbcRfRNdUt/B7w00Iu34TruV9W8Qf4UAr9FAVZXL80zdTRqqfy8ltYz9/Ct/h+hFSuWMYh/rHfvlx/f/AwAA//8D
AFBLAwQUAAYACAAAACEABW/ErbMAAADJAAAAGAAoAGN1c3RvbVhtbC9pdGVMUHJvcHM1LnhtbCCiJAAooCAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjcsKwjAQRfe//xBmX5ViMorYYYB9wpuuQzrVQjMjTRRB/Hezui+4Z7v
G8cYoDk4FFyoUAgee4Guhu4XmyhQcTkqHMjExoghv1uPtt2se5ccjHxhOeEQeRiyHpuDHzbY6l1tLV64LW6oAA//8DAFBLAwQU
AAYACAAAACEA2jZYA7YAAADJAAAAGAoAGN1c3RvbVhtl/aXHMzLnhtbCCiJAAooCAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAMjU0LwjAQRO+C/yHsPSbgrypGabEF7wpeQrMrTRRB/O/nNMw8mLc/fsIg3jjGns
nCfKZBIHlue7pbuF4aWYCTYHVrBia0QzHw3Syb+OudcnFxCQewEwaRhz7n+WThW5i63mzqI/ViWcjlqmrk1phaVtu60ro
sy8asfyCymvJNtPBI6blTKvoHBhdn/ETKsOMxuJTreFfcdb3HE/tXQErKaL1W/pX14RYGUIc/AAAA//8DAFBLAwQUAAYA
CAAAACEA2UvLGLGAAADJAAAAGAoAGN1c3RvbVhtbC9pdGVMUHJvcHMyLnhtbCCiJAAooCAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAMjcGKwjAURfcD8w/h7YqjqVjbgvsZmG1IX22heU9d4D9xxEtYXhHEHkxc55zs7sq4NKbt23Rh9rHV
JgthoEkudhppuF359vhHm5D4g06nwfxCOeEwaRhz7n+a0/9u3nf/WqE7TyEtXXXWvyZ+WTfvvM94FRTW+6o+Pvpbu5+q5HE
V6g+IrKZEgBy3cU3pslIr+jsHFJT+QMh/7jlNr7N4gBcWwAA//8DAFBLAWIAtABQABgAIAAAAIQAez7Q/7QEAAAFYAAATAAAAAAAAAAAAAAAAAAAAAAAABbQ29udGVudF9UeXBlc1
0ueG1sUEsBAi0AFAAGAAgAAAAhABNevmUFAQAA3wIAAAsAAAAAAAAAAAAAAAAAAAJgQAAF9yZWxzLy5yZWxzUEsBAi0AFAA
GAAgAAAhAOjfpD5wAQAAPwcAABoAAAAAAAAAAAAAAAAAAXAcAAHhsL19yZWxzL3dvcmtib29rLnhtbC5yZWxzUEsBAi0A
FAAGAAgAAAhAMCDPT6FAQAAxQIAAA8AAAAAAAAAAAAAAAAADAoAAHhsL3dvcmtib29rLnhtbFBLAQItABQABgAIAAAAI
QAallvEaQEAAAIwCAAAAAAAAAAAAAAAAAAL4LAAB4bC93b3Jrc2hlZXRzL3NoZWV0MS54bWxQSwECLQAUAAYACAAAACAAACC
EEAN4tbid4AAADZAQAAIwAAAAAAAAAAAAAAAAABdDQAAeGwvdGhlbWUvdGhlbWUxLnhtbFBLAQItABQABgAIAAAAI
QAEATU8QsE8IAAABCQAAIwAAAAAAAAAAAAAAAAAAABkCAAAeGwvdGhlbWUvdGhlbWUxLnhtbFBLAQItABQABgAIAAAAI
QAEANKEJksIAAABCQAAIwAAAAAAAAAAAAAAAAAAAB/DwAAeGwvdGhlbWUvdGhlbWUxLnhtbFBLAQItABQABgAIAAAAI
QAEACkEM6sYAAABHAQAAKQAAAAAAAAAAAAAAAAACCEAAAeGwvG

l2b3RUYWJsZXMvX3JlbHMvcGl2b3RUYWJsZTEueG1sLnJlbHNQSwECLQAUAAYACAAAACEAojvW8o4EAAA5QQAAHgAAAAA
AAAAAAAAAACPEQAAeGwvcGl2b3RUYWJsZXMvcGl2b3RUYWJsZTEueG1sUEsBAi0AFAAGAAgAAAAhAERJdzrnAAAAigEA
ABQAAAAAAAAAAAAAAAAAWRYAAHhsL3NoYXJlZFN0cmluZ3MueG1sUEsBAi0AFAAGAAgAAAAhABclP8r0BgAATCMAABMAA
AAAAAAAAAAAAAAAchcAAHhsL3RoZW1lL3RoZW1lMS54bWxQSwECLQAUAAYACAAAACEAs6QaM6YCAACYBgAAAAAAAAAAAA
AAAAAAACXHgAAeGwvd29ya3NoZWV0cy9zaGVldDEueG1sUEsBAi0AFAAGAAgAAAAhAJQj1tyoAgAANgYAAA0AAAAAAAAA
AAAAAAAAACyEAAHhsL3N0eWxlcy54bWxQSwECLQAUAAYACAAAACEAGpZbxGkBAACMAgAAEAAAAAAAAAAAAAAAAABGJAAA
eGwvd29ya3NoZWV0cy9zaGVldDMueG1sUEsBAi0AFAAGAAgAAAAhAJCVG1V8JwAARooBACcAAAAAAAAAAAAAAAAA5SUAA
HhsL3Bpdm90Q2FjbGUvcGl2b3RDYWNoZUvcGl2b3RDYWNoZURlZmluaXRpb24xLnhtbFBLAQItABQABgAIAAAAIQAIKrfqzEAAANgCAAASAA
AAAAAAAAAAAAAAAAKZNAAB4bC9jb25uZWN0aW9uc54bWxQSwECLQAUAAYACAAAACEALmnZ2SkBAAA0BAAAJwAAAAAAAAA
AAAAAAACiTwAAeGwvcHJpbnRlclNldHRpbmdzL3ByaW50ZXJTZXR0aW5nczMuYmluUEsBAi0AFAAGAAgAAAAhAC5p2dkp
AQAANQAACcAAAAAAAAAAAAAAAAAEFEAAHhsL3ByaW50ZXJTZXR0aW5ncy9wcmludGVyU2V0dGluZ3MyLmJpblBLAQItA
BQABgAIAAAAIQAuadnZKQEAADQEAAAnAAAAAAAAAAAAAAAAAAH5SAAB4bC9wcmludGVyU2V0dGluZ3MvcHJpbnRlclNldH
RpbmdzMS5iaW5QSwECLQAUAAYACAAAACEAvYRiI5AAAADbAAAAEwAAAAAAAAAAAAAAAAAADsUwAAY3VzdG9tWG1sL2l0ZW0
zLnhtbFBLAQItABQABgAIAAAAIQAMxBqSwwAAACgBAAAeAAAAAAAAAAAAAAAAAANVUAABjdXN0b21YbWwvX3JlbHMvaXRl
bTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEAe/MCo8MAAAAoAQAAHgAAAAAAAAAAAAAAAAADUVQAAY3VzdG9tWG1sL19yZ
WxzL2l0ZW0zLnhtbC5yZWxzUEsBAi0AFAAGAAgAAAAhAFyWJyLDAAAAKAEAAB4AAAAAAAAAAAAAAAAAA01YAAGN1c3RvbV
htbC9fcmVscy9pdGVtMi54bWwucmVsc1BLAQItABQABgAIAAAAIQB0Pzl6wgAAACgBAAAeAAAAAAAAAAAAAAAAAANJXAAB
jdXN0b21YbWvaXRlbTQueG1sLnJlbHNQSwECLQAUAAYACAAAACEABk/l30sBAAgAAEQAAAAAAAAAAAAAAAAAAAAAA
AADQWAAAZG9jUHJvcHMvY29yZS54bWxQSwECLQAUAAYACAAAACEAkFZEkpQBAABHAwAAEAAAAAAAAAAAAAAAAAAABSWwAAZ
G9jUHJvcHMvYXBwLnhtbFBLAQItABQABgAIAAAAIQAOjUVeIwEAAAcCAAATAAAAAAAAAAAAAAAAAABxeAABBkb2NQcm9wcy
9jdXN0b20ueG1sUEsBAi0AFAAGAAgAAAAhAF5X0aBVAQAAZQYAABMAAAAAAAAAAAAAAAAAeGAAAGN1c3RvbVhtbC9pdGGV
tMi54bWxQSwECLQAUAAYACAAAACEAgGM8eBMJAAA5KQAAEwAAAAAAAAAAAAAAAAAAmYgAAY3VzdG9tWG1sL2l0ZW0xLnht
bFBLAQItABQABgAIAAAAIQAFb8StswAAAAkAAAAYAAAAAAAAAAAAAAAAAAJJrAABjdXN0b21YbWvaXRlbVByb3BzNC54b
WxQSwECLQAUAAYACAAAACEA2jZGA7YAAADJAAAAGAAAAAAAAAAAAAAAAAACjbAAAY3VzdG9tWG1sL2l0ZW0Qcm9wczMueG
1sUEsBAi0AFAAGAAgAAAAhANlLyxi4AAAAyQAAABgAAAAAAAAAAAAAAAAAt20AAGN1c3RvbVhtbC9pdGVtUHJvcHMyLnh
tbFBLAQItABQABgAIAAAAIQA3E8qItwAAAMkAAAAYAAAAAAAAAAAAAAAAAM1uAABjdXN0b21YbWvaXRlbVByb3BzMS54
bWxQSwECLQAUAAYACAAAACEAkswz0hgBAAAkAgAAEwAAAAAAAAAAAAAAAADibwAAY3VzdG9tWG1sL2l0ZW00LnhtbFBLB
QYAAAAIwAjAMMJAABTcQAAAA==</GetWorkbookResult>
      </GetWorkbookResponse>
    </soap:Body>
</soap:Envelope>
```

▪ Close the workbook session on the protocol server.

### Request

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

Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CloseWorkbookResponse
xmlns="http://schemas.microsoft.com/office/excel/server/webservices" />
  </soap:Body>
```

```
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

There are security risks associated with exposing session identifiers (2). If the session identifier is exposed, it is possible for an attacker to read information from, or modify data in, a session (2) on the protocol server. An implementer of this protocol is to consider keeping session identifiers protected. There could be some cases where it is desirable to expose the session identifier, but implementers are to use caution in how they expose session identifiers and consider the security risks.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

```xml
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/office/excel/server/webservices"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="GetApiVersion">
        <xs:complexType />
      </xs:element>
      <xs:element name="GetApiVersionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="GetApiVersionResult"
type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="ArrayOfStatus">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Status" nillable="true"
type="tns:Status" />
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="Status">
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="Name" type="xs:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="Severity" type="xs:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="Message" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
      <xs:element name="NewWorkbook">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="path" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"
/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NewWorkbookResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="NewWorkbookResult" type="xs:string"
/>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
```

```xml
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OpenWorkbookForEditing">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"
 />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OpenWorkbookForEditingResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookForEditingResult"
 type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OpenWorkbook">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="uiCultureName" type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="dataCultureName" type="xs:string"
 />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OpenWorkbookResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="OpenWorkbookResult"
 type="xs:string" />
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CloseWorkbook">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="CloseWorkbookResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Refresh">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
```

```
                <xs:element minOccurs="0" maxOccurs="1" name="connectionName" type="xs:string" />
              </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="RefreshResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="SaveWorkbook">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="SaveWorkbookResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="SaveWorkbookCopy">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="workbookPath" type="xs:string" />
              <xs:element minOccurs="1" maxOccurs="1" name="workbookType"
  type="tns:WorkbookType" />
              <xs:element minOccurs="1" maxOccurs="1" name="saveOptions" type="tns:SaveOptions"
  />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:simpleType name="WorkbookType">
          <xs:restriction base="xs:string">
            <xs:enumeration value="FullWorkbook" />
            <xs:enumeration value="FullSnapshot" />
            <xs:enumeration value="PublishedItemsSnapshot" />
          </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="SaveOptions">
          <xs:list>
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="None" />
                <xs:enumeration value="AllowOverwrite" />
              </xs:restriction>
            </xs:simpleType>
          </xs:list>
        </xs:simpleType>
        <xs:element name="SaveWorkbookCopyResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
```

```
          </xs:complexType>
        </xs:element>
        <xs:element name="Calculate">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
  type="tns:RangeCoordinates" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:complexType name="RangeCoordinates">
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="Row" type="xs:int" />
            <xs:element minOccurs="1" maxOccurs="1" name="Column" type="xs:int" />
            <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int" />
            <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int" />
          </xs:sequence>
        </xs:complexType>
        <xs:element name="CalculateResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CalculateA1">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CalculateA1Response">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetRange">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="rangeCoordinates"
  type="tns:RangeCoordinates" />
              <xs:element minOccurs="1" maxOccurs="1" name="formatted" type="xs:boolean" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetRangeResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="GetRangeResult"
  type="tns:ArrayOfAnyType" />
```

[MS-ESP] — v20120410
Excel Web Services Protocol Specification

Copyright © 2012 Microsoft Corporation.

Release: Tuesday, April 10, 2012

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

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

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

```
              </xs:complexType>
            </xs:element>
            <xs:element name="SetRangeA1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="rangeValues"
        type="tns:ArrayOfAnyType" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="SetRangeA1Response">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="SetCell">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
                  <xs:element minOccurs="1" maxOccurs="1" name="row" type="xs:int" />
                  <xs:element minOccurs="1" maxOccurs="1" name="column" type="xs:int" />
                  <xs:element minOccurs="0" maxOccurs="1" name="cellValue" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="SetCellResponse">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="SetCellA1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="sheetName" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="rangeName" type="xs:string" />
                  <xs:element minOccurs="0" maxOccurs="1" name="cellValue" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="SetCellA1Response">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="CalculateWorkbook">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
```

```
              <xs:element minOccurs="1" maxOccurs="1" name="calculateType"
type="tns:CalculateType" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:simpleType name="CalculateType">
          <xs:restriction base="xs:string">
            <xs:enumeration value="Recalculate" />
            <xs:enumeration value="CalculateFull" />
          </xs:restriction>
        </xs:simpleType>
        <xs:element name="CalculateWorkbookResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetSessionInformation">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetSessionInformationResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="serverVersion" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="uiCultureNameUsed" type="xs:string"
/>
              <xs:element minOccurs="0" maxOccurs="1" name="dataCultureNameUsed"
type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CancelRequest">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="CancelRequestResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetWorkbook">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="1" maxOccurs="1" name="workbookType"
type="tns:WorkbookType" />
            </xs:sequence>
          </xs:complexType>
```

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

```
        </xs:simpleType>
        <xs:element name="SetCalculationOptionsResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="GetChartImageUrl">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="sessionId" type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="size" type="tns:Size" />
              <xs:element minOccurs="0" maxOccurs="1" name="chartName" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:complexType name="Size">
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="Width" type="xs:int" />
            <xs:element minOccurs="1" maxOccurs="1" name="Height" type="xs:int" />
          </xs:sequence>
        </xs:complexType>
        <xs:element name="GetChartImageUrlResponse">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="1" name="GetChartImageUrlResult"
  type="xs:string" />
              <xs:element minOccurs="0" maxOccurs="1" name="status" type="tns:ArrayOfStatus" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:simpleType name="RangeReference">
          <xs:restriction base="xs:string" />
        </xs:simpleType>
        <xs:simpleType name="SheetName">
          <xs:restriction base="xs:string" />
        </xs:simpleType>
      </xs:schema>
    </wsdl:types>
    <wsdl:message name="GetApiVersionSoapIn">
      <wsdl:part name="parameters" element="tns:GetApiVersion" />
    </wsdl:message>
    <wsdl:message name="GetApiVersionSoapOut">
      <wsdl:part name="parameters" element="tns:GetApiVersionResponse" />
    </wsdl:message>
    <wsdl:message name="NewWorkbookSoapIn">
      <wsdl:part name="parameters" element="tns:NewWorkbook" />
    </wsdl:message>
    <wsdl:message name="NewWorkbookSoapOut">
      <wsdl:part name="parameters" element="tns:NewWorkbookResponse" />
    </wsdl:message>
    <wsdl:message name="OpenWorkbookForEditingSoapIn">
      <wsdl:part name="parameters" element="tns:OpenWorkbookForEditing" />
    </wsdl:message>
    <wsdl:message name="OpenWorkbookForEditingSoapOut">
      <wsdl:part name="parameters" element="tns:OpenWorkbookForEditingResponse" />
    </wsdl:message>
    <wsdl:message name="OpenWorkbookSoapIn">
```

```
    <wsdl:part name="parameters" element="tns:OpenWorkbook" />
  </wsdl:message>
  <wsdl:message name="OpenWorkbookSoapOut">
    <wsdl:part name="parameters" element="tns:OpenWorkbookResponse" />
  </wsdl:message>
  <wsdl:message name="CloseWorkbookSoapIn">
    <wsdl:part name="parameters" element="tns:CloseWorkbook" />
  </wsdl:message>
  <wsdl:message name="CloseWorkbookSoapOut">
    <wsdl:part name="parameters" element="tns:CloseWorkbookResponse" />
  </wsdl:message>
  <wsdl:message name="RefreshSoapIn">
    <wsdl:part name="parameters" element="tns:Refresh" />
  </wsdl:message>
  <wsdl:message name="RefreshSoapOut">
    <wsdl:part name="parameters" element="tns:RefreshResponse" />
  </wsdl:message>
  <wsdl:message name="SaveWorkbookSoapIn">
    <wsdl:part name="parameters" element="tns:SaveWorkbook" />
  </wsdl:message>
  <wsdl:message name="SaveWorkbookSoapOut">
    <wsdl:part name="parameters" element="tns:SaveWorkbookResponse" />
  </wsdl:message>
  <wsdl:message name="SaveWorkbookCopySoapIn">
    <wsdl:part name="parameters" element="tns:SaveWorkbookCopy" />
  </wsdl:message>
  <wsdl:message name="SaveWorkbookCopySoapOut">
    <wsdl:part name="parameters" element="tns:SaveWorkbookCopyResponse" />
  </wsdl:message>
  <wsdl:message name="CalculateSoapIn">
    <wsdl:part name="parameters" element="tns:Calculate" />
  </wsdl:message>
  <wsdl:message name="CalculateSoapOut">
    <wsdl:part name="parameters" element="tns:CalculateResponse" />
  </wsdl:message>
  <wsdl:message name="CalculateA1SoapIn">
    <wsdl:part name="parameters" element="tns:CalculateA1" />
  </wsdl:message>
  <wsdl:message name="CalculateA1SoapOut">
    <wsdl:part name="parameters" element="tns:CalculateA1Response" />
  </wsdl:message>
  <wsdl:message name="GetRangeSoapIn">
    <wsdl:part name="parameters" element="tns:GetRange" />
  </wsdl:message>
  <wsdl:message name="GetRangeSoapOut">
    <wsdl:part name="parameters" element="tns:GetRangeResponse" />
  </wsdl:message>
  <wsdl:message name="GetRangeA1SoapIn">
    <wsdl:part name="parameters" element="tns:GetRangeA1" />
  </wsdl:message>
  <wsdl:message name="GetRangeA1SoapOut">
    <wsdl:part name="parameters" element="tns:GetRangeA1Response" />
  </wsdl:message>
  <wsdl:message name="GetCellSoapIn">
    <wsdl:part name="parameters" element="tns:GetCell" />
  </wsdl:message>
  <wsdl:message name="GetCellSoapOut">
    <wsdl:part name="parameters" element="tns:GetCellResponse" />
  </wsdl:message>
```

```
<wsdl:message name="GetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:GetCellA1" />
</wsdl:message>
<wsdl:message name="GetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:GetCellA1Response" />
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNames" />
</wsdl:message>
<wsdl:message name="GetPublishedItemNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetPublishedItemNamesResponse" />
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapIn">
  <wsdl:part name="parameters" element="tns:GetSheetNames" />
</wsdl:message>
<wsdl:message name="GetSheetNamesSoapOut">
  <wsdl:part name="parameters" element="tns:GetSheetNamesResponse" />
</wsdl:message>
<wsdl:message name="SetRangeSoapIn">
  <wsdl:part name="parameters" element="tns:SetRange" />
</wsdl:message>
<wsdl:message name="SetRangeSoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeResponse" />
</wsdl:message>
<wsdl:message name="SetRangeA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetRangeA1" />
</wsdl:message>
<wsdl:message name="SetRangeA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetRangeA1Response" />
</wsdl:message>
<wsdl:message name="SetCellSoapIn">
  <wsdl:part name="parameters" element="tns:SetCell" />
</wsdl:message>
<wsdl:message name="SetCellSoapOut">
  <wsdl:part name="parameters" element="tns:SetCellResponse" />
</wsdl:message>
<wsdl:message name="SetCellA1SoapIn">
  <wsdl:part name="parameters" element="tns:SetCellA1" />
</wsdl:message>
<wsdl:message name="SetCellA1SoapOut">
  <wsdl:part name="parameters" element="tns:SetCellA1Response" />
</wsdl:message>
<wsdl:message name="CalculateWorkbookSoapIn">
  <wsdl:part name="parameters" element="tns:CalculateWorkbook" />
</wsdl:message>
<wsdl:message name="CalculateWorkbookSoapOut">
  <wsdl:part name="parameters" element="tns:CalculateWorkbookResponse" />
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapIn">
  <wsdl:part name="parameters" element="tns:GetSessionInformation" />
</wsdl:message>
<wsdl:message name="GetSessionInformationSoapOut">
  <wsdl:part name="parameters" element="tns:GetSessionInformationResponse" />
</wsdl:message>
<wsdl:message name="CancelRequestSoapIn">
  <wsdl:part name="parameters" element="tns:CancelRequest" />
</wsdl:message>
<wsdl:message name="CancelRequestSoapOut">
  <wsdl:part name="parameters" element="tns:CancelRequestResponse" />
```

```
    </wsdl:message>
    <wsdl:message name="GetWorkbookSoapIn">
      <wsdl:part name="parameters" element="tns:GetWorkbook" />
    </wsdl:message>
    <wsdl:message name="GetWorkbookSoapOut">
      <wsdl:part name="parameters" element="tns:GetWorkbookResponse" />
    </wsdl:message>
    <wsdl:message name="SetParametersSoapIn">
      <wsdl:part name="parameters" element="tns:SetParameters" />
    </wsdl:message>
    <wsdl:message name="SetParametersSoapOut">
      <wsdl:part name="parameters" element="tns:SetParametersResponse" />
    </wsdl:message>
    <wsdl:message name="SetCalculationOptionsSoapIn">
      <wsdl:part name="parameters" element="tns:SetCalculationOptions" />
    </wsdl:message>
    <wsdl:message name="SetCalculationOptionsSoapOut">
      <wsdl:part name="parameters" element="tns:SetCalculationOptionsResponse" />
    </wsdl:message>
    <wsdl:message name="GetChartImageUrlSoapIn">
      <wsdl:part name="parameters" element="tns:GetChartImageUrl" />
    </wsdl:message>
    <wsdl:message name="GetChartImageUrlSoapOut">
      <wsdl:part name="parameters" element="tns:GetChartImageUrlResponse" />
    </wsdl:message>
    <wsdl:portType name="ExcelServiceSoap">
      <wsdl:operation name="GetApiVersion">
        <wsdl:input message="tns:GetApiVersionSoapIn" />
        <wsdl:output message="tns:GetApiVersionSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="NewWorkbook">
        <wsdl:input message="tns:NewWorkbookSoapIn" />
        <wsdl:output message="tns:NewWorkbookSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="OpenWorkbookForEditing">
        <wsdl:input message="tns:OpenWorkbookForEditingSoapIn" />
        <wsdl:output message="tns:OpenWorkbookForEditingSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="OpenWorkbook">
        <wsdl:input message="tns:OpenWorkbookSoapIn" />
        <wsdl:output message="tns:OpenWorkbookSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="CloseWorkbook">
        <wsdl:input message="tns:CloseWorkbookSoapIn" />
        <wsdl:output message="tns:CloseWorkbookSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="Refresh">
        <wsdl:input message="tns:RefreshSoapIn" />
        <wsdl:output message="tns:RefreshSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbook">
        <wsdl:input message="tns:SaveWorkbookSoapIn" />
        <wsdl:output message="tns:SaveWorkbookSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbookCopy">
        <wsdl:input message="tns:SaveWorkbookCopySoapIn" />
        <wsdl:output message="tns:SaveWorkbookCopySoapOut" />
      </wsdl:operation>
      <wsdl:operation name="Calculate">
```

```
          <wsdl:input message="tns:CalculateSoapIn" />
          <wsdl:output message="tns:CalculateSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="CalculateA1">
          <wsdl:input message="tns:CalculateA1SoapIn" />
          <wsdl:output message="tns:CalculateA1SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetRange">
          <wsdl:input message="tns:GetRangeSoapIn" />
          <wsdl:output message="tns:GetRangeSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetRangeA1">
          <wsdl:input message="tns:GetRangeA1SoapIn" />
          <wsdl:output message="tns:GetRangeA1SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetCell">
          <wsdl:input message="tns:GetCellSoapIn" />
          <wsdl:output message="tns:GetCellSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetCellA1">
          <wsdl:input message="tns:GetCellA1SoapIn" />
          <wsdl:output message="tns:GetCellA1SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetPublishedItemNames">
          <wsdl:input message="tns:GetPublishedItemNamesSoapIn" />
          <wsdl:output message="tns:GetPublishedItemNamesSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetSheetNames">
          <wsdl:input message="tns:GetSheetNamesSoapIn" />
          <wsdl:output message="tns:GetSheetNamesSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SetRange">
          <wsdl:input message="tns:SetRangeSoapIn" />
          <wsdl:output message="tns:SetRangeSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SetRangeA1">
          <wsdl:input message="tns:SetRangeA1SoapIn" />
          <wsdl:output message="tns:SetRangeA1SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SetCell">
          <wsdl:input message="tns:SetCellSoapIn" />
          <wsdl:output message="tns:SetCellSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="SetCellA1">
          <wsdl:input message="tns:SetCellA1SoapIn" />
          <wsdl:output message="tns:SetCellA1SoapOut" />
        </wsdl:operation>
        <wsdl:operation name="CalculateWorkbook">
          <wsdl:input message="tns:CalculateWorkbookSoapIn" />
          <wsdl:output message="tns:CalculateWorkbookSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="GetSessionInformation">
          <wsdl:input message="tns:GetSessionInformationSoapIn" />
          <wsdl:output message="tns:GetSessionInformationSoapOut" />
        </wsdl:operation>
        <wsdl:operation name="CancelRequest">
          <wsdl:input message="tns:CancelRequestSoapIn" />
          <wsdl:output message="tns:CancelRequestSoapOut" />
        </wsdl:operation>
```

```
      <wsdl:operation name="GetWorkbook">
        <wsdl:input message="tns:GetWorkbookSoapIn" />
        <wsdl:output message="tns:GetWorkbookSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="SetParameters">
        <wsdl:input message="tns:SetParametersSoapIn" />
        <wsdl:output message="tns:SetParametersSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="SetCalculationOptions">
        <wsdl:input message="tns:SetCalculationOptionsSoapIn" />
        <wsdl:output message="tns:SetCalculationOptionsSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="GetChartImageUrl">
        <wsdl:input message="tns:GetChartImageUrlSoapIn" />
        <wsdl:output message="tns:GetChartImageUrlSoapOut" />
      </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExcelServiceSoap" type="tns:ExcelServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetApiVersion">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetApiVersion"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="NewWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/NewWorkbook"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbookForEditing">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookForEditi
ng" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OpenWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
```

```
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CloseWorkbook">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CloseWorkbook"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="Refresh">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Refresh"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbook">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbook"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbookCopy">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopy"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="Calculate">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Calculate"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CalculateA1">
```

```
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateA1"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRange">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRange"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRangeA1">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRangeA1"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCell">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCell"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCellA1">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCellA1"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetPublishedItemNames">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetPublishedItemName
s" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
```

```
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSheetNames">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetRange">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRange"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetRangeA1">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetCell">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCell"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetCellA1">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
```

```
      <wsdl:operation name="CalculateWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateWorkbook"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSessionInformation">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformatio
n" style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CancelRequest">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CancelRequest"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetWorkbook">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetParameters">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetParameters"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetCalculationOptions">
      <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCalculationOption
s" style="document" />
      <wsdl:input>
```

```
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetChartImageUrl">
        <soap:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
style="document" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="ExcelServiceSoap12" type="tns:ExcelServiceSoap">
      <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="GetApiVersion">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetApiVersion"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="NewWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/NewWorkbook"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="OpenWorkbookForEditing">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbookForEditi
ng" style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="OpenWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/OpenWorkbook"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
```

```
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CloseWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CloseWorkbook"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="Refresh">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Refresh"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbook">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbook"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SaveWorkbookCopy">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SaveWorkbookCopy"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="Calculate">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/Calculate"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="CalculateA1">
```

```
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateA1"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRange">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRange"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetRangeA1">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetRangeA1"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCell">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCell"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCellA1">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetCellA1"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetPublishedItemNames">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetPublishedItemName
s" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
```

```
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="GetSheetNames">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSheetNames"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SetRange">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRange"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SetRangeA1">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetRangeA1"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SetCell">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCell"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="SetCellA1">
        <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCellA1"
style="document" />
        <wsdl:input>
          <soap12:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap12:body use="literal" />
        </wsdl:output>
      </wsdl:operation>
```

```
      <wsdl:operation name="CalculateWorkbook">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CalculateWorkbook"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetSessionInformation">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetSessionInformatio
n" style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CancelRequest">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/CancelRequest"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetWorkbook">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetWorkbook"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetParameters">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetParameters"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetCalculationOptions">
      <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/SetCalculationOption
s" style="document" />
      <wsdl:input>
```

```
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetChartImageUrl">
    <soap12:operation
soapAction="http://schemas.microsoft.com/office/excel/server/webservices/GetChartImageUrl"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

7   Appendix B: Product Behavior

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

- Microsoft® Office SharePoint® Server 2007

- Microsoft® SharePoint® Server 2010

- Microsoft® SharePoint® Server 15 Technical Preview

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

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

<1> Section 2.2.9.2: Excel Services 2010 can use any of the values specified in section 2.2.9.1.

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

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

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

<5> Section 3.1.4: Group 1 is supported in Excel Services 2007 and Excel Services 2010. Group 2 is supported in Excel Services 2010.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

<25> Section 3.1.4.26: Excel Services 2007 and Excel Services 2010 set the values sent by the protocol client and do not return an error if **rangeValues** contains values in only one dimension (a single column (2) or a single row (2)), and if the first worksheet cell of **rangeCoordinates** falls on a workbook parameter.

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

<27> Section 3.1.4.27: Excel Services 2007 and Excel Services 2010 set the values sent by the protocol client and do not return an error if **rangeValues** contains values in only one dimension (a single column or a single row), and if the first worksheet cell of **rangeCoordinates** falls on a workbook parameter.

8 Change Tracking

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

9   Index

Release: Tuesday, April 10, 2012