[MS-OMWWH]:

Office Mobile Word Web Handler Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial Availability
8/28/2009	0.2	Editorial	Revised and edited the technical content
11/6/2009	0.3	Editorial	Revised and edited the technical content
2/19/2010	1.0	Major	Updated and revised the technical content
3/31/2010	1.01	Editorial	Revised and edited the technical content
4/30/2010	1.02	Editorial	Revised and edited the technical content
6/7/2010	1.03	Editorial	Revised and edited the technical content
6/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.05	Major	Significantly changed the technical content.
9/27/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.05	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	1.6	Minor	Clarified the meaning of the technical content.
4/11/2012	1.6	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	1.7	Minor	Clarified the meaning of the technical content.
10/8/2012	1.7	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	1.7	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.7	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.7	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.7	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.8	Minor	Clarified the meaning of the technical content.
7/31/2014	1.8	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
10/30/2014	1.8	None	No changes to the meaning, language, or formatting of the technical content.
6/23/2016	1.8	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	1.8	None	No changes to the meaning, language, or formatting of the technical content.
9/19/2017	1.9	Minor	Clarified the meaning of the technical content.
12/12/2017	2.0	Major	Significantly changed the technical content.
6/19/2018	2.1	Minor	Clarified the meaning of the technical content.
12/11/2018	2.2	Minor	Clarified the meaning of the technical content.
6/18/2019	2.2	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Office Mobile Word Web Handler protocol retrieves document display information from the protocol server for a document that is optimized for viewing on a mobile device.

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

1.1 Glossary

This document uses the following terms:

absolute URL: The full Internet address of a page or other World Wide Web resource. The absolute URL includes a protocol, such as "http," a network location, and an optional path and file name — for example, http://www.treyresearch.net/.

document: An object in a content database such as a file, folder, list, or **site**. Each object is identified by a **URI**.

endnote: A note that appears at the end of a section or document and that is referenced by text in the main body of the document. An endnote consists of two linked parts, a reference mark within the main body of text and the corresponding text of the note.

footnote: A note that appears at the end of a page, section, chapter, or publication. It explains, comments on, or provides references for text in the main body of a document. A footnote consists of two linked parts, a reference mark within the main body of the document and the corresponding text of the note.

front-end web server: A server that hosts webpages, performs processing tasks, and accepts requests from protocol clients and sends them to the appropriate back-end server for further processing.

header row: A row in a table, typically the first row, that contains labels for columns in the table.

HTTP GET: An HTTP method for retrieving a resource, as described in [RFC2616].

hyperlink: A relationship between two anchors, as described in [RFC1866].

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

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

point: A unit of measurement for fonts and spacing. A point is equal to 1/72 of an inch.

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL supports server and, optionally, client authentication using X.509 certificates [X509] and [RFC5280]. SSL is superseded by Transport Layer Security (TLS). TLS version 1.0 is based on SSL version 3.0 [SSL3].

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

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

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

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

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

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

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

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

1.2 References

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

1.2.1 Normative References

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

[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

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

[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/

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

1.2.2 Informative References

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

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

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

[SOAP1.2-2/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 2: Adjuncts (Second Edition)", W3C Recommendation, April 2007, http://www.w3.org/TR/2007/REC-soap12-part2-20070427

1.3 Overview

This protocol specifies the communication between the client and the **front-end Web server** to obtain the contents of a document<1> stored on the server in a form suitable to be displayed on mobile devices.

All communication is transported over **Hypertext Transfer Protocol (HTTP)**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

Each method is an **HTTP GET** request, as described in [RFC2616], that accepts a set of parameters and returns an HTTP response depending upon the method invoked. The parameters of the method are sent as query parameters as part of the **URL**, as described in [RFC2616] section 9.1.

1.4 Relationship to Other Protocols

This protocol uses **HTTP**, as described in [RFC2616], or **HTTPS**, as described in [RFC2818].

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

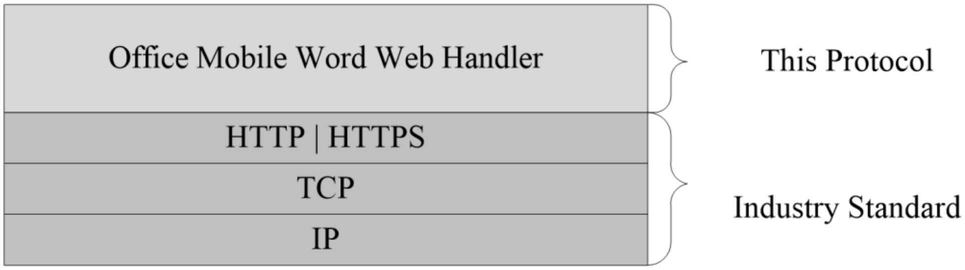

Figure 1: This protocol in relation to other protocols

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 to send requests for the document metadata is formed by appending "/_layouts/MobileDocHandler.ashx" to the URL of the site, for example: http://www.contoso.com/Repository/_layouts/MobileDocHandler.ashx.

The protocol server endpoint to send requests for a page image is formed by appending the value of the **url** attribute of the **CT_PageImage** element (section 3.1.4.1.1.4.4) contained in the metadata of the requested document to the URL of the site, for example: http://www.contoso.com/Repository/_layouts/MobilePageHandler.ashx.

The protocol server endpoint to send requests for page information is formed by appending the value of the **url** attribute of the **CT_PageXml** element (section 3.1.4.1.1.4.3) contained in the metadata of the requested document to the URL of the site, for example: http://www.contoso.com/Repository/_layouts/MobilePageHandler.ashx.

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

1.6 Applicability Statement

This protocol is a precursor to the **SOAP** protocol, as described in [SOAP1.1], [SOAP1.2-1/2007], and [SOAP1.2-2/2007], and can be used in similar situations.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses **HTTP** or **HTTPS** as transport for the **HTTP GET** methods. The HTTP headers used are discussed in the following individual protocol sections.

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
(none)	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2]

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

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

2.2.9 Common Data Structures

This specification does not define any common **XML schema** data structures.

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. 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 MUST interpret HTTP status codes returned by the protocol server as specified in [RFC2616] section 10.

3.1 Server Details

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

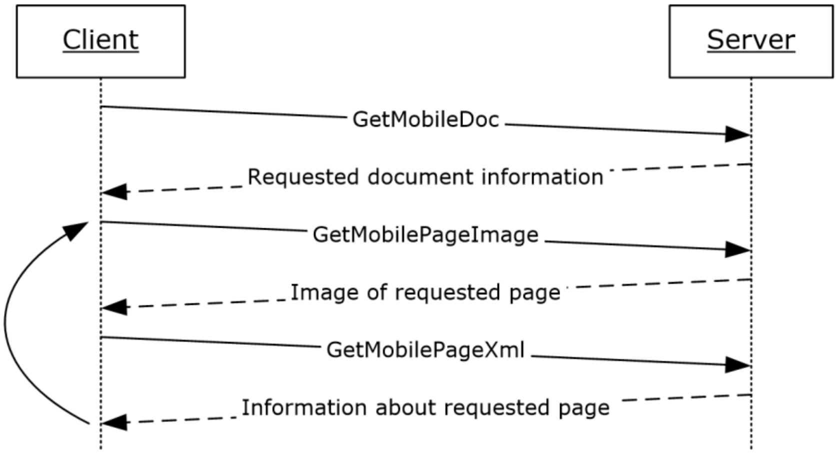

Figure 2: Sample communication between protocol client and protocol server

First, a protocol client calls **GetMobileDoc** (section 3.1.4.1.1), and the protocol server responds with the information about the requested document.

The protocol client, using the previous information, makes one or more calls to **GetMobilePageImage** (section 3.1.4.2.1) and **GetMobilePageXml** (section 3.1.4.2.2), and the protocol server responds with the image of the requested page and information about the requested page, respectively.

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.

3.1.2 Timers

None.

3.1.3 Initialization

The protocol server MUST expose its Web methods at a **URL**, which builds upon a base URL.

The URL MUST be constructed as base URL/_layouts/handler name.

The base URL can be any web site URL, such as "http://www.contoso.com/Repository". The part of _layouts/handler Name can be either of the following:

```
    _layouts/MobileDocHandler.ashx
    _layouts/MobilePageHandler.ashx
```

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
GetMobileDoc	Retrieves the metadata of a document.
GetMobilePageImage	Retrieves the image of a page in a document.
GetMobilePageXml	Retrieves information about content of a page in a document.
GetLatestDocumentVersion	Retrieves the current version of a document.

3.1.4.1 MobileDocHandler

The methods in this section are supported by MobileDocHandler.ashx.

3.1.4.1.1 GetMobileDoc

This method retrieves the metadata of a document. The **URL** for this method is constructed by appending "_layouts/MobileDocHandler.ashx" to the **site** name. The parameters to be passed while calling this method are as follows:

doc: A string element ([XMLSCHEMA2/2] section 3.2.1) that specifies the **absolute URL** of the document.

If any of the preceding conditions for the parameters are not satisfied, the method returns one of the error values as specified in section 3.1.4.1.1.1.

3.1.4.1.1.1 Return Values

This operation sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table.

Value	Description
200	Success. The protocol server returns XML specifying the document metadata, which contains a **mobileDoc** (section 3.1.4.1.1.3.1) root element. An **X-DocVersion** HTTP header is present, which is an xsd:string ([XMLSCHEMA2/2] section 3.2.1) that specifies the version of the document. An **X-DocUrl** HTTP header is present, which is an xsd:string ([XMLSCHEMA2/2] section 3.2.1) that specifies the **URL** of the document.

Value	Description
503	Failure. The server also sets an HTTP header **X-Error xsd:unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.
404	Failure.

3.1.4.1.1.2 Messages

None.

3.1.4.1.1.3 Elements

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

Element	Description
mobileDoc	A **CT_MobileDoc** element (section 3.1.4.1.1.4.7) that specifies the metadata of the document.

3.1.4.1.1.3.1 mobileDoc

A **CT_MobileDoc** element (section 3.1.4.1.1.4.7) that specifies the metadata of the document.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="mobileDoc" type="CT_MobileDoc"/>
```

3.1.4.1.1.4 Complex Types

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

Complex type	Description
CT_PageSet	A page set in the document.
CT_Document	The metadata of the document.
CT_PageXml	The information required to retrieve the information about pages of the document.
CT_PageImage	The information required to obtain the image of a page in the document.
CT_DocData	Metadata of the document.
CT_Ignorable	Reserved. MUST be ignored by client.

Complex type	Description
CT_MobileDoc	A root element that specifies the metadata of the document.

3.1.4.1.1.4.1 CT_PageSet

Referenced by: CT_Document

This complex type specifies a page set in the document. A page set is a set of contiguous pages that have the same width and height.

Attributes:

width: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the width of a page in the current page set in terms of dots. The width of the page in inches MUST be set to **width** divided by **dxpInch**, as specified in **CT_Document** (section 3.1.4.1.1.4.2).

height: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the height of a page in the current page set in terms of dots. The height of the page, in inches, MUST be set to **height** divided by **dypInch**, as specified in **CT_Document** (section 3.1.4.1.1.4.2).

count: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the number of pages in the page set.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_PageSet">
  <xsd:attribute name="width" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="height" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="count" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

3.1.4.1.1.4.2 CT_Document

Referenced by: CT_DocData, CT_MobileDoc

This complex type contains the metadata of the document.

Child Elements:

pageset: A **CT_PageSet** element (section 3.1.4.1.1.4.1) that specifies a set of contiguous pages that have the same width and height. The **CT_PageSet** (section 3.1.4.1.1.4.1) elements MUST occur in the order in which the page sets themselves occur in the **document**.

Attributes:

pages: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the number of pageset elements in the CT_Document type element.

dxpInch: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the number of dots per inch in the horizontal direction of the document.

dypInch: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the number of dots per inch in the vertical direction of the document.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Document">
  <xsd:sequence>
    <xsd:element name="pageset" type="CT_PageSet" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="pages" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="dxpInch" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="dypInch" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

3.1.4.1.1.4.3 CT_PageXml

Referenced by: CT_MobileDoc

This complex type specifies the information required to retrieve the information about pages of the document.

Attributes:

url: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the relative **URL** of the **MobilePageHandler** (section 3.1.4.2).

page: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the name of the parameter that defines the page range for which information is requested. The value of this attribute MUST be set to **n**. For more details on **n**, see **GetMobilePageXml** (section 3.1.4.2.2). The parameter is combined with **url** to construct the complete request URL to obtain the page information.

start: This attribute MUST be ignored by client on receipt.

length: This attribute MUST be ignored by client on receipt.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_PageXml">
  <xsd:attribute name="url" type="xsd:string" use="required"/>
  <xsd:attribute name="page" type="xsd:string" use="required"/>
  <xsd:attribute name="start" type="xsd:string" use="required"/>
  <xsd:attribute name="length" type="xsd:string" use="required"/>
</xsd:complexType>
```

3.1.4.1.1.4.4 CT_PageImage

Referenced by: CT_MobileDoc

This complex type specifies the information required to obtain the image of a page in the document.

Attributes:

url: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the relative **URL** of the **MobilePageHandler** (section 3.1.4.2).

image: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the name of the parameter that defines the page for which the image is requested. The value of this attribute MUST be set to **n**. For more details on **n**, see **GetMobilePageImage** (section 3.1.4.2.1). This parameter is combined with **url** to construct the complete request URL to obtain the page images.

start: This attribute MUST be ignored by client on receipt.

length: This attribute MUST be ignored by client on receipt.

width: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the name of the parameter that defines the width of the image requested. The value of this attribute MUST be set to **width**. This parameter is combined with **url** to construct the complete request URL to obtain the page images.

height: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the name of the parameter that defines the height of the image requested. The value of this attribute MUST be set to **height**. This parameter is combined with **url** to construct the complete request URL to obtain the page images.

format: A **string** attribute ([XMLSCHEMA2/2] section 3.2.1) that specifies the name of the parameter that defines the format of the image requested. The value of this attribute MUST be set to **fmt**. This parameter is combined with **url** to construct the complete request URL to obtain the page images.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_PageImage">
  <xsd:attribute name="url" type="xsd:string" use="required"/>
  <xsd:attribute name="image" type="xsd:string" use="required"/>
  <xsd:attribute name="start" type="xsd:string" use="required"/>
  <xsd:attribute name="length" type="xsd:string" use="required"/>
  <xsd:attribute name="width" type="xsd:string" use="required"/>
  <xsd:attribute name="height" type="xsd:string" use="required"/>
  <xsd:attribute name="format" type="xsd:string" use="required"/>
</xsd:complexType>
```

3.1.4.1.1.4.5 CT_DocData

Referenced by: CT_MobileDoc

This complex type contains metadata of the document.

Child Elements:

document: A **CT_Document** element (section 3.1.4.1.1.4.2) that specifies the metadata of the document.

manifest: A **CT_Ignorable** element (section 3.1.4.1.1.4.6) that is reserved. It MUST be ignored by client.

status: Reserved. MUST be ignored by client.

dialog: A **CT_Ignorable** element that is reserved. It MUST be ignored by client.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_DocData">
  <xsd:sequence>
    <xsd:element name="document" type="CT_Document" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="manifest" type="CT_Ignorable" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="status" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="dialog" type="CT_Ignorable" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.1.1.4.6 CT_Ignorable

Referenced by: CT_DocData, CT_Page

Reserved. MUST be ignored by client.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Ignorable">
  <xsd:sequence>
    <xsd:any processContents="skip"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.1.1.4.7 CT_MobileDoc

Referenced by: mobileDoc

This complex type is a root element and specifies the metadata of the document. When CT_MobileDoc is returned by using **PrepareThumbnail**, it MUST have a **docdata** child element. Otherwise, it MUST have a **document** child element. A CT_MobileDoc element MUST NOT have both **docdata** and **document** as child elements.

Child Elements:

docdata: A **CT_DocData** element (section 3.1.4.1.1.4.5) that specifies information about the document. This child MUST be present if **document** is not present.

document: A **CT_Document** element (section 3.1.4.1.1.4.2) that specifies the document metadata. This child MUST be present if **docdata** is not present.

pageXml: A **CT_PageXml** element (section 3.1.4.1.1.4.3) that specifies information required to obtain information for pages of the **document**.

pageImage: A **CT_PageImage** element (section 3.1.4.1.1.4.4) that specifies information required to obtain images of the pages of the document.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_MobileDoc">
  <xsd:sequence>
    <xsd:element name="docdata" type="CT_DocData" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="document" type="CT_Document" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="pageXml" type="CT_PageXml" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="pageImage" type="CT_PageImage" minOccurs="1" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.1.1.5 Simple Types

None.

3.1.4.1.1.6 Attributes

None.

3.1.4.1.1.7 Groups

None.

3.1.4.1.1.8 Attribute Groups

None.

3.1.4.1.2 GetLatestDocumentVersion

This method retrieves the current version of document. The **URL** for this method is constructed by appending "_layouts/MobileDocHandler.ashx" to the **site** name. The parameters to be passed while calling this method are as follows:

doc: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the absolute URL of the document.

getCurrentDocumentVersion: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies that the current version of the document is to be returned. The value of this parameter MUST be set to **true**, or this parameter is ignored on receipt. When this parameter is ignored on reception, the return value is 200.

If any of the preceding conditions for the parameters are not satisfied, this method returns one of the error values as specified in section 3.1.4.1.2.1.

3.1.4.1.2.1 Return Values

This method sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table.

Value	Description
200	Success. The protocol server sets an **X-DocVersion** HTTP header that is an **xsd:string** ([XMLSCHEMA2/2] section 3.2.1) that specifies the version of the document. An **X-DocUrl** HTTP header is present, which is an xsd:string ([XMLSCHEMA2/2] section 3.2.1) that specifies the **URL** of the document.
503	Failure. The server also sets an HTTP header **X-Error xsd:unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.
404	Failure.

3.1.4.1.3 PrepareThumbnail

This method starts the preparation of the thumbnail of the first page of a document. The **URL** for this method is constructed by appending "_layouts/MobileDocHandler.ashx" to the **site** name. The parameters to be passed while calling this method are as follows:

doc: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the **absolute URL** of the document.

type: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies that the thumbnail of the first page of the document is to be prepared. The value of this parameter MUST be set to **thumbnail**, or this parameter is ignored on receipt.

If any of the preceding conditions for the parameters are not satisfied, the method returns one of the error values specified in section 3.1.4.1.3.1.

3.1.4.1.3.1 Return Values

This operation sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table:

Value	Description
200	**Success:** The protocol server returns XML specifying the document metadata, which contains a **mobileDoc** (section 3.1.4.1.1.3.1) root element. An **X-DocVersion** HTTP header is present, which is an xsd:string ([XMLSCHEMA2/2] section 3.2.1) that specifies the version of the document. An **X-DocUrl** HTTP header is present, which is an xsd:string ([XMLSCHEMA2/2] section 3.2.1) that specifies the **URL** of the document.
503	**Failure:** The server also sets an HTTP header **X-Error xsd:unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.

3.1.4.2 MobilePageHandler

The methods in this section are supported by MobilePageHandler.ashx.

3.1.4.2.1 GetMobilePageImage

This method obtains images of pages in a document. The **URL** for this method is constructed by appending the value of the **url** attribute of the **CT_PageImage** (section 3.1.4.1.1.4.4) element contained in the metadata of the requested document to the URL of the **site**. The URL of the site MUST first have "/_layouts/" appended to it. The parameters to be passed while calling this method are as follows:

n: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the page image to be obtained. To request the image of page 1, **n** MUST be set to **p1.img**. To request the image of page 2, **n** MUST be set to **p2.img**. The value of **n** MUST be of the form **p*k*.img**, where ***k*** is an integer greater than or equal to 1 and less than or equal to the number of pages in the document. The number of pages in the document is calculated as the sum of the **count** attribute of all **CT_PageSet** (section 3.1.4.1.1.4.1) elements.

width: An **unsignedInt** element ([XMLSCHEMA2/2]) section 3.3.22) that specifies the width of the requested image, in pixels. This parameter is optional. The default value of this parameter is 480. The maximum permissible value of this parameter is 1500.

height: An **unsignedInt** element ([XMLSCHEMA2/2] section 3.3.22) that specifies the height of the requested image, in pixels. This parameter is optional. The default value of this parameter is 621. The maximum permissible value of this parameter is 2000.

fmt: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the image format of the requested image. Image formats other than **png** and **jpeg** are encoded to **jpeg** by default.

When one of these two parameters, **width** and **height** is sent with a valid value, and the other one is not sent or sent with an empty value or sent with value 0, or both parameters are sent with value 0, the image returned has the default values of width and height.

When the **fmt** parameter is not sent or sent with an empty value or sent with an invalid value, the image returned has the default value of fmt.

If any of the preceding conditions for the parameters are not satisfied, the method returns one of the error values specified in section 3.1.4.2.1.1.

3.1.4.2.1.1 Return Values

This method sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table.

Value	Description
200	**Success:** The protocol server returns an image of the requested page in the specified format.
503	**Failure:** The server also sets an HTTP header **X-Error** xsd:unsignedInt ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.
404	**Failure.**

3.1.4.2.2 GetMobilePageXml

This method obtains information about the content of pages in a document. The **URL** for this method is constructed by appending the value of the **url** attribute of the **CT_PageXml** (section 3.1.4.1.1.4.3) element contained in the metadata of the requested document to the URL of the **site**. The URL of the site MUST first have "/_layouts/" appended to it. The parameters to be passed while calling this method are as follows:

n: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the page range for which the information is to be obtained. To request the information for the first 10 pages, **n** MUST be set to **p_1_10.xml**. To request the information for the next 10 pages, **n** MUST be set to **p_11_20.xml**. The value of **n** MUST be of the form **p_*i*_*j*.xml**, where ***i* = *10*(k-1) + 1*** and ***j* = *10*k***, where ***k*** is an integer greater than or equal to 1 and less than or equal to 1/10 of the number of pages in the document, rounded off to the next integer. The number of pages in the document is calculated as the sum of the **count** attribute of all **CT_PageSet** (section 3.1.4.1.1.4.1) elements.

If any of the preceding conditions for the parameters are not satisfied, the method returns of the error values as specified in section 3.1.4.2.2.1.

3.1.4.2.2.1 Return Values

This method sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table.

Value	Description
200	**Success:** The protocol server returns XML specifying information about the requested page range, which contains a **Pages** (section 3.1.4.2.2.3.1) root element.

Value	Description
503	**Failure:** The server also sets an HTTP header **X-Error** xsd:unsignedInt ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.
404	**Failure.**

3.1.4.2.2.2 Messages

None.

3.1.4.2.2.3 Elements

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

Element	Description
Pages	A **CT_Pages** element (section 3.1.4.2.2.4.16) that specifies the content of pages requested by this operation.

3.1.4.2.2.3.1 Pages

This element is a **CT_Pages** element (section 3.1.4.2.2.4.16) that specifies the content of pages requested by this operation.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this element.

```
<xsd:element name="Pages" type="CT_Pages"/>
```

3.1.4.2.2.4 Complex Types

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

Complex type	Description
CT_TextLine	A line of text.
CT_LinkTargetInternal	An internal **hyperlink** destination.
CT_Link	A hyperlink.
CT_Paragraph	A paragraph.
CT_Image	The type and position of an image.
CT_EndNote	An endnote.

Complex type	Description
CT_FootNote	A footnote.
CT_TableOfContents	The table of contents of a document.
CT_TextBox	A textbox.
CT_List	A list.
CT_TableDataCell	A data cell in a table row, as specified in **CT_TableRow** (section 3.1.4.2.2.4.13).
CT_TableHeaderCell	A cell in a table **header row**, as specified by **CT_TableRow** (section 3.1.4.2.2.4.13).
CT_TableRow	A row in a table, as specified in **CT_Table** (section 3.1.4.2.2.4.14).
CT_Table	A table.
CT_Page	A page.
CT_Pages	The set of pages in a document.

3.1.4.2.2.4.1 CT_TextLine

Referenced by: CT_Paragraph, CT_Page

This complex type specifies a line of text.

Attributes:

id: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies a unique identifier of the text line. This value MUST be unique across all text lines that occur in the document, and MUST be ordered in sequence, based on the order in which the text lines appear in the document. This value MUST be zero-indexed.

l: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the left-most position of the text line, in **points** relative to the top left of the page on which it occurs. This value MUST be greater than or equal to zero and less than the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

t: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the top-most position of the text line, in points relative to the top left of the page on which it occurs. This value MUST be greater than or equal to zero and less than the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

w: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the width of the text line, in points. This value MUST be greater than zero.

h: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the height of the text, in points. This value MUST be greater than zero.

b: An **unsignedByte** attribute ([XMLSCHEMA2/2] section 3.3.24) that specifies that a line break follows the text line. This value MUST be set to 1 when a new line follows the text line. Otherwise, this attribute MUST be omitted.

a: An **unsignedShort** attribute ([XMLSCHEMA2/2] section 3.3.23) that specifies the angle of vertical text. This value MUST be set to 90 or 270. If the text is horizontal, this value MUST be omitted. The possible values are described in the following table.

Value	Description
90	The text is rotated 90 degrees clockwise from the horizontal.
270	The text is rotated 270 degrees clockwise from the horizontal.

rtl: An **unsignedByte** attribute ([XMLSCHEMA2/2] section 3.3.24) that specifies that the text flow direction is right to left. This value MUST be 1 when the text flow is right to left. Otherwise, this attribute MUST be omitted.

s: An **unsignedByte** attribute ([XMLSCHEMA2/2] section 3.3.24) that specifies that the text line is stretched or condensed horizontally. The value MUST be 1 to represent that the text line is changed horizontally either by stretching or condensing. Otherwise, this attribute MUST be omitted.

type: An **ST_TextLineType** attribute (section 3.1.4.2.2.5.3) that specifies the text line type.

gr: An **unsignedByte** attribute ([XMLSCHEMA2/2] section 3.3.24) that specifies that the glyphs in the text line are rotated. This value MUST be set to 1 when the text line glyphs are rotated**.** Otherwise, this attribute MUST be omitted.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TextLine">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
      <xsd:attribute name="l" type="xsd:float" use="required"/>
      <xsd:attribute name="t" type="xsd:float" use="required"/>
      <xsd:attribute name="w" type="xsd:float" use="required"/>
      <xsd:attribute name="h" type="xsd:float" use="required"/>
      <xsd:attribute name="b" type="xsd:unsignedByte" use="optional"/>
      <xsd:attribute name="a" type="xsd:unsignedShort" use="optional"/>
      <xsd:attribute name="rtl" type="xsd:unsignedByte" use="optional"/>
      <xsd:attribute name="s" type="xsd:unsignedByte" use="optional"/>
      <xsd:attribute name="type" type="ST_TextLineType" use="optional"/>
      <xsd:attribute name="gr" type="xsd:unsignedByte" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

3.1.4.2.2.4.2 CT_LinkTargetInternal

Referenced by: CT_Link

This complex type specifies an internal **hyperlink** destination. The destination specifies the target page and target position relative to the top left of the page.

Attributes:

p: An **unsignedInt** attribute([XMLSCHEMA2/2] section 3.3.22) that specifies the target page number. This value MUST be greater than or equal to 1 and less than or equal to the total number of pages in the document.

l: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the left-most position of the hyperlink destination, in **points** relative to the left of the target page. This value MUST be greater than or equal to zero and less than the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

t: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the top position of the hyperlink destination, in points relative to the top of the target page. This value MUST be greater than or equal to zero and less than the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_LinkTargetInternal">
  <xsd:attribute name="p" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="l" type="xsd:float" use="required"/>
  <xsd:attribute name="t" type="xsd:float" use="required"/>
</xsd:complexType>
```

3.1.4.2.2.4.3 CT_Link

Referenced by: CT_Paragraph, CT_Page

This complex type specifies a **hyperlink**. The attributes **l**, **t**, **w**, and **h** specify the source rectangle relative to the top left of the page on which it occurs.

If the hyperlink source element for the link is of type **CT_TextLine** (section 3.1.4.2.2.4.1), the attributes **s** and **n** specify the location of the source text in the following text line. The CT_Link element MUST appear before the **CT_TextLine** (section 3.1.4.2.2.4.1) element of the text line or another **CT_Link** element that occurs in the same text line. Hyperlink sources that occur in the same text line MUST NOT overlap.

If the hyperlink source element for the link is of type **CT_Image** (section 3.1.4.2.2.4.5), the attributes **s** and **n** MUST be zero, and the **CT_Link** element MUST appear before the **CT_Image** (section 3.1.4.2.2.4.5) element.

If the destination of the hyperlink is located in the document, it MUST have a child of type **CT_LinkTargetInternal** (section 3.1.4.2.2.4.2). Otherwise, the text of the **CT_Link** (section 3.1.4.2.2.4.3) element MUST be the destination **URI**. The **CT_Link** (section 3.1.4.2.2.4.3) element MUST NOT have both a child of type **CT_LinkTargetInternal** (section 3.1.4.2.2.4.2) and text.

Child Elements:

LT: A **CT_LinkTargetInternal** element (section 3.1.4.2.2.4.2) that specifies an internal hyperlink destination.

Attributes:

l: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the left-most position of the hyperlink source rectangle, in **points** relative to the top left of the page on which it occurs. This value MUST be greater than or equal to zero and less than the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

t: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the top position of the hyperlink source rectangle, in points relative to the top left of the page on which it occurs. This value MUST be greater than or equal to zero and less than the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

w: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the width of the hyperlink source rectangle, in points relative to the top left of the page on which it occurs. This value MUST be greater than zero and less than or equal to the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

h: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the height of the hyperlink source rectangle, in points relative to the top left of the page on which it occurs. This value MUST be greater than zero and less than or equal to the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

s: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the starting character of the hyperlink source text. When the hyperlink source contains text from a text line, as specified by **CT_TextLine** (section 3.1.4.2.2.4.1), **s** specifies the index of the first character of the hyperlink source in the Unicode text. The first character in the Unicode text line is zero-indexed. This value MUST be greater than or equal to zero and MUST be less than the total number of characters in the text line, minus 1. If the hyperlink source does not contain text, **s** MUST be zero.

n: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the number of characters that the hyperlink source text spans in the text line, as specified by **CT_TextLine** (section 3.1.4.2.2.4.1). This value MUST be greater than or equal to zero and less than or equal to the number of characters in the text line, minus **s**. If the hyperlink source does not contain text, **n** MUST be set to zero.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Link" mixed="true">
  <xsd:choice maxOccurs="1" minOccurs="0">
    <xsd:element name="LT" type="CT_LinkTargetInternal"/>
  </xsd:choice>
  <xsd:attribute name="l" type="xsd:float" use="required"/>
  <xsd:attribute name="t" type="xsd:float" use="required"/>
  <xsd:attribute name="w" type="xsd:float" use="required"/>
  <xsd:attribute name="h" type="xsd:float" use="required"/>
  <xsd:attribute name="s" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="n" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

3.1.4.2.2.4.4 CT_Paragraph

Referenced by: CT_Paragraph, CT_EndNote, CT_TableDataCell, CT_List, CT_TableHeaderCell, CT_FootNote, CT_TableOfContents, CT_TextBox, CT_Page

This complex type specifies a paragraph.

Child Elements:

T: A **CT_TextLine** element (section 3.1.4.2.2.4.1) that specifies a text line in a paragraph.

L: A **CT_Link** element (section 3.1.4.2.2.4.3) that specifies a link in a paragraph.

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a paragraph.

Attributes:

id: This attribute MUST be ignored by client on receipt.

storyId: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the unique identifier of the text flow in which this paragraph occurs. Each story represents a distinct text flow in the document.

type: An **ST_ParagraphType** attribute (section 3.1.4.2.2.5.2) that specifies the type of paragraph.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Paragraph">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="T" type="CT_TextLine"/>
    <xsd:element name="L" type="CT_Link"/>
    <xsd:element name="P" type="CT_Paragraph"/>
  </xsd:choice>
  <xsd:attribute name="id" type="xsd:unsignedInt" use="optional"/>
  <xsd:attribute name="storyId" type="xsd:unsignedInt" use="required"/>
  <xsd:attribute name="type" type="ST_ParagraphType" use="optional"/>
</xsd:complexType>
```

3.1.4.2.2.4.5 CT_Image

Referenced by: CT_EndNote, CT_FootNote, CT_Page

This complex type specifies the type and position of an image.

Attributes:

type: An **ST_ImageType** attribute (section 3.1.4.2.2.5.1) that specifies the image type.

l: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the left-most position of the image, in **points** relative to the top left of the page on which it occurs. This value MUST be less than the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

t: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the top position of the image, in points relative to the top left of the page on which it occurs. This value MUST be greater than or equal to zero and less than the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

w: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the width of the image, in points. This value MUST be greater than zero and less than the width of the page, as specified by **width** of **CT_PageSet** (section 3.1.4.1.1.4.1).

h: A **float** attribute ([XMLSCHEMA2/2] section 3.2.4) that specifies the height of the image, in points. This value MUST be greater than zero and less than the height of the page, as specified by **height** of **CT_PageSet** (section 3.1.4.1.1.4.1).

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Image">
  <xsd:attribute name="type" type="ST_ImageType" use="required"/>
  <xsd:attribute name="l" type="xsd:float" use="required"/>
  <xsd:attribute name="t" type="xsd:float" use="required"/>
  <xsd:attribute name="w" type="xsd:float" use="required"/>
  <xsd:attribute name="h" type="xsd:float" use="required"/>
</xsd:complexType>
```

3.1.4.2.2.4.6 CT_EndNote

Referenced by: CT_Page

This complex type specifies an **endnote**.

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in an endnote.

Table: A **CT_Table** element (section 3.1.4.2.2.4.14) that specifies a table in an endnote.

Image: A **CT_Image** element (section 3.1.4.2.2.4.5) that specifies an image in an endnote.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in an endnote.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_EndNote">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="P" type="CT_Paragraph"/>
    <xsd:element name="Table" type="CT_Table"/>
    <xsd:element name="Image" type="CT_Image"/>
    <xsd:element name="List" type="CT_List"/>
  </xsd:choice>
</xsd:complexType>
```

3.1.4.2.2.4.7 CT_FootNote

Referenced by: CT_Page

This complex type specifies a **footnote**.

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a footnote.

Image: A **CT_Image** element (section 3.1.4.2.2.4.5) that specifies an image in a footnote.

Table: A **CT_Table** element (section 3.1.4.2.2.4.14) that specifies a table in a footnote.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in a footnote.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_FootNote">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="P" type="CT_Paragraph"/>
    <xsd:element name="Image" type="CT_Image"/>
    <xsd:element name="Table" type="CT_Table"/>
    <xsd:element name="List" type="CT_List"/>
  </xsd:choice>
</xsd:complexType>
```

3.1.4.2.2.4.8 CT_TableOfContents

Referenced by: CT_Page

This complex type specifies the table of contents of a document.

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a table of contents.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TableOfContents">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" name="P" type="CT_Paragraph"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.2.2.4.9 CT_TextBox

Referenced by: CT_Page

This complex type specifies a text box.

Child Elements:

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in a text box.

P: A **CT_Paragraph** (section 3.1.4.2.2.4.4) element that specifies a paragraph in a text box.

Table: A **CT_Table** element (section 3.1.4.2.2.4.14) that specifies a table in a text box.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TextBox">
  <xsd:choice maxOccurs="unbounded" minOccurs="0">
    <xsd:element name="List" type="CT_List"/>
    <xsd:element name="P" type="CT_Paragraph"/>
    <xsd:element name="Table" type="CT_Table"/>
  </xsd:choice>
</xsd:complexType>
```

3.1.4.2.2.4.10 CT_List

Referenced by: CT_TableDataCell, CT_List, CT_TableHeaderCell, CT_EndNote, CT_FootNote, CT_TextBox, CT_Page

This complex type specifies a list.

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a list.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a sub-list.

Attributes:

i: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the level of indent of the list. This value MUST be greater or equal to zero. This value is zero-based. For each level of indent, the value increases by one.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_List">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```
      <xsd:element name="P" type="CT_Paragraph"/>
      <xsd:element name="List" type="CT_List"/>
    </xsd:choice>
    <xsd:attribute name="i" type="xsd:unsignedInt" use="required"/>
  </xsd:complexType>
```

3.1.4.2.2.4.11 CT_TableDataCell

Referenced by: CT_TableRow

This complex type specifies a data cell in a table row, as specified by **CT_TableRow** (section 3.1.4.2.2.4.13).

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a data cell.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in a data cell.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
  <xsd:complexType name="CT_TableDataCell">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="P" type="CT_Paragraph"/>
      <xsd:element name="List" type="CT_List"/>
    </xsd:choice>
  </xsd:complexType>
```

3.1.4.2.2.4.12 CT_TableHeaderCell

Referenced by: CT_TableRow

This complex type specifies a cell in a table **header row**, as specified by **CT_TableRow** (section 3.1.4.2.2.4.13).

Child Elements:

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a header cell.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in a header cell.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
  <xsd:complexType name="CT_TableHeaderCell">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="P" type="CT_Paragraph"/>
      <xsd:element name="List" type="CT_List"/>
    </xsd:choice>
  </xsd:complexType>
```

3.1.4.2.2.4.13 CT_TableRow

Referenced by: CT_Table

This complex type specifies a row in a table, as specified by **CT_Table** (section 3.1.4.2.2.4.14).

Child Elements:

TD: A **CT_TableDataCell** element (section 3.1.4.2.2.4.11) that specifies a table cell in a row if the row is not a **header row**.

TH: A **CT_TableHeaderCell** element (section 3.1.4.2.2.4.12) that specifies a table cell in a row if the row is a header row.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_TableRow">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TD" type="CT_TableDataCell"/>
    <xsd:element name="TH" type="CT_TableHeaderCell"/>
  </xsd:choice>
</xsd:complexType>
```

3.1.4.2.2.4.14 CT_Table

Referenced by: CT_EndNote, CT_FootNote, CT_TextBox, CT_Page

This complex type specifies a table.

Child Elements:

TR: A **CT_TableRow** element (section 3.1.4.2.2.4.13) that specifies a row in a table.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Table">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" name="TR" type="CT_TableRow"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.2.2.4.15 CT_Page

Referenced by: CT_Pages

This complex type specifies a page.

Child Elements:

Table: A **CT_Table** element (section 3.1.4.2.2.4.14) that specifies a table in a page.

P: A **CT_Paragraph** element (section 3.1.4.2.2.4.4) that specifies a paragraph in a page.

Image: A **CT_Image** element (section 3.1.4.2.2.4.5) that specifies an image in a page.

Header: A **CT_Ignorable** element (section 3.1.4.1.1.4.6) that is reserved. It MUST be ignored by client.

Footer: A **CT_Ignorable** element that is reserved. It MUST be ignored by client.

TOC: A **CT_TableOfContents** element (section 3.1.4.2.2.4.8) that specifies a table of contents in a page.

TextBox: A **CT_TextBox** element (section 3.1.4.2.2.4.9) that specifies a text box in a page.

List: A **CT_List** element (section 3.1.4.2.2.4.10) that specifies a list in a page.

Footnote: A **CT_FootNote** element (section 3.1.4.2.2.4.7) that specifies a **footnote** in a page.

Endnote: A **CT_EndNote** element (section 3.1.4.2.2.4.6) that specifies an **endnote** in a page.

T: A **CT_TextLine** element (section 3.1.4.2.2.4.1) that MUST be ignored by client.

L: A **CT_Link** element (section 3.1.4.2.2.4.3) that specifies a link in a page.

Attributes:

id: An **unsignedInt** attribute ([XMLSCHEMA2/2] section 3.3.22) that specifies the page number. Page numbers MUST be unique across all elements of type **CT_Page** (section 3.1.4.2.2.4.15) in a document. Pages numbers MUST be in ascending order. The value MUST be greater than or equal to 1 and less than or equal to the number of pages in the document.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Page">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Table" type="CT_Table"/>
    <xsd:element name="P" type="CT_Paragraph"/>
    <xsd:element name="Image" type="CT_Image"/>
    <xsd:element name="Header" type="CT_Ignorable"/>
    <xsd:element name="Footer" type="CT_Ignorable"/>
    <xsd:element name="TOC" type="CT_TableOfContents"/>
    <xsd:element name="TextBox" type="CT_TextBox"/>
    <xsd:element name="List" type="CT_List"/>
    <xsd:element name="Footnote" type="CT_FootNote"/>
    <xsd:element name="Endnote" type="CT_EndNote"/>
    <xsd:element name="T" type="CT_TextLine"/>
    <xsd:element name="L" type="CT_Link"/>
  </xsd:choice>
  <xsd:attribute name="id" type="xsd:unsignedInt" use="required"/>
</xsd:complexType>
```

3.1.4.2.2.4.16 CT_Pages

Referenced by: Pages

This complex type specifies the set of pages in a document.

Child Elements:

Page: A **CT_Page** element (section 3.1.4.2.2.4.15) that specifies a page in a document.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this complex type.

```
<xsd:complexType name="CT_Pages">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" name="Page" type="CT_Page"/>
  </xsd:sequence>
</xsd:complexType>
```

3.1.4.2.2.5 Simple Types

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

Simple type	Description
ST_ImageType	The type of an image, as specified in **CT_Image** (section 3.1.4.2.2.4.5).
ST_ParagraphType	The type of a paragraph, as specified in **CT_Paragraph** (section 3.1.4.2.2.4.4).
ST_TextLineType	The type of a text line, as specified in **CT_TextLine** (section 3.1.4.2.2.4.1).

3.1.4.2.2.5.1 ST_ImageType

Referenced by: CT_Image

This simple type specifies the type of an image, as specified by **CT_Image** (section 3.1.4.2.2.4.5).

Value	Meaning
Figure	The image is a figure.
Chart	The image is a chart.
Diagram	The image is a diagram.
Unknown	The image is none of the previous.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_ImageType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Figure"/>
    <xsd:enumeration value="Chart"/>
    <xsd:enumeration value="Diagram"/>
    <xsd:enumeration value="Unknown"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.1.4.2.2.5.2 ST_ParagraphType

Referenced by: CT_Paragraph

This simple type specifies the type of a paragraph, as specified by **CT_Paragraph** (section 3.1.4.2.2.4.4).

Value	Meaning
Heading	The paragraph uses a heading style.
ListBody	The paragraph occurs in a list.
TOCI	The paragraph text occurs in a table of contents.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_ParagraphType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Heading"/>
    <xsd:enumeration value="ListBody"/>
    <xsd:enumeration value="TOCI"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.1.4.2.2.5.3 ST_TextLineType

Referenced by: CT_TextLine

This simple type specifies the type of a text line, as specified by **CT_TextLine** (section 3.1.4.2.2.4.1).

Value	Meaning
InlineShape	The text line contains an inline object.

The following W3C XML Schema ([XMLSCHEMA1/2] section 2.1) fragment specifies the contents of this simple type.

```
<xsd:simpleType name="ST_TextLineType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="InlineShape"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.1.4.2.2.6 Attributes

None.

3.1.4.2.3 GetThumbnail

This method obtains the thumbnail of the first page of a document. The **URL** for this method is constructed by appending the value of the **url** attribute of the **CT_PageImage** (section 3.1.4.1.1.4.4) element contained in the metadata of the requested document returned by **PrepareThumbnail** to the URL of the **site**. The URL of the site MUST first have "/_layouts/" appended to it. The parameters to be passed while calling this method are as follows:

n: A **string** element ([XMLSCHEMA2/2] section 3.2.1) that specifies the thumbnail to be obtained. The value of this parameter MUST be set to **t1.jpg** or this parameter is ignored on receipt.

If any of the preceding conditions for the parameters are not satisfied, the method returns one of the error values as specified in section 3.1.4.2.3.1.

3.1.4.2.3.1 Return Values

This method sends an HTTP response back to the client, and the HTTP response status code MUST be one of the values in the following table.

Value	Description
200	**Success:** The protocol server returns a thumbnail of the first page of the document.
503	**Failure:** The server also sets an HTTP header **X-Error xsd:unsignedInt** ([XMLSCHEMA2/2] section 3.3.22) in the response. If the value is greater than or equal to 100 and less than 200, the protocol client SHOULD retry this call; otherwise, this header MUST be ignored.
404	**Failure.**

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following examples contain a sample interaction between the protocol client and the protocol server. These examples are illustrative of the preceding specification, and they do not cover all possible structure usage scenarios. They are not intended to replace the preceding specification, but rather to clarify and enhance it. In the following examples, the client is attempting to view a document located at http://sptestamd/testdocs/Example%20Document.docx.

4.1 Document Information

The **URL** for **GetMobileDoc** (section 3.1.4.1.1) is constructed as follows:

```
http://sptestamd/_layouts/MobileDocHandler.ashx?doc=http://sptestamd/testdocs/Example%20Docum
ent.docx
```

The protocol client then performs an **HTTP GET** on this URL.

The protocol server responds with a **CT_MobileDoc** (section 3.1.4.1.1.4.7) that contains information about the document.

```
<?xml version="1.0" encoding="utf-8"?>
<mobileDoc>
  <document pages="1" dxpInch="294912" dypInch="294912">
    <pageset width="983040" height="1474560" count="4" />
  </document>
  <pageXml
url="./MobilePageHandler.ashx?d=H%3Asptestamd%2F%2Ftestdocs%2FExample%20Document%2E
docx&amp;z=B15D182E-D3F6-4383-AC4A-23EB2D3C64C92&amp;v=00000000-0000-0000-0000-
00000000010b" page="n" start="s" length="l" />
  <pageImage
url="./MobilePageHandler.ashx?d=H%3Asptestamd%2F%2Ftestdocs%2FExample%20Document%2E
docx&amp;z=B15D182E-D3F6-4383-AC4A-23EB2D3C64C92&amp;v=00000000-0000-0000-0000-
00000000010b" image="n" start="s" length="l" width="width" height="height"
format="fmt" />
</mobileDoc>
```

The contained **pageXml** and **pageImage** elements specify information about how to obtain the page information and page images, respectively.

4.2 Page Image

The **pageImage** element of the **CT_MobileDoc** (section 3.1.4.1.1.4.7) specifies how to obtain the page images.

The **URL** to obtain the image of the first page is constructed as follows:

```
http://sptestamd/_layouts/MobilePageHandler.ashx?d=H%3Asptestamd%2F%2Ftestdocs%2FExample%20Do
cument%2Edocx&z=B15D182E-D3F6-4383-AC4A-23EB2D3C64C92&v=00000000-0000-0000-0000-
00000000010b&n=p1.img
```

The protocol client performs an **HTTP GET** on this URL to obtain the default-sized image of the first page.

4.3 Page Information

The **pageXml** element of the **CT_MobileDoc** (section 3.1.4.1.1.4.7) specifies how to obtain page information.

The **URL** to obtain information about pages 1 to 10 is as follows:

```
http://sptestamd/_layouts/MobilePageHandler.ashx?d=H%3Asptestamd%2F%2Ftestdocs%2FExample%20Do
cument%2Edocx&z=B15D182E-D3F6-4383-AC4A-23EB2D3C64C92&v=00000000-0000-0000-0000-
00000000010b&n=p_1_10.xml
```

The protocol client performs an **HTTP GET** on this URL.

The protocol server responds with a **CT_Pages** complex type (section 3.1.4.2.2.4.16) that contains information about the requested page range.

```
<?xml version="1.0"?>
<Pages>
  <Page id="1">
    <Header>
      <P storyId="2">
        <T id="0" l="0" t="2.9" w="47.2" h="13" b="1">Header </T>
      </P>
    </Header>
    <Footer>
      <P storyId="3">
        <T id="1" l="0" t="467.8" w="43.1" h="13" b="1">Footer </T>
      </P>
    </Footer>
    <P storyId="1">
      <L l="-3" t="16.3" w="195.2" h="32.1" s="0" n="26">
        <LT p="2" l="-3" t="16.3"></LT>
      </L>
      <T id="2" l="0" t="19.9" w="192.6" h="13" b="1">DOCMAPBEGIN:00:DOCMAPBEGIN
</T>
    </P>
    <P storyId="1">
      <L l="-3" t="48.3" w="147.2" h="32.1" s="0" n="21">
        <LT p="2" l="-3" t="160.6"></LT>
      </L>
      <T id="3" l="0" t="52" w="149.5" h="13" b="1">DOCMAPITEM:1:Header 1 </T>
    </P>
    <P storyId="1">
      <L l="-3" t="80.4" w="146.7" h="32.1" s="0" n="21">
        <LT p="3" l="0" t="0"></LT>
      </L>
      <T id="4" l="0" t="84" w="145.5" h="13" b="1">DOCMAPITEM:1:Contents </T>
    </P>
    <P storyId="1">
      <L l="-3" t="112.4" w="365.6" h="32.1" s="0" n="23">
        <LT p="3" l="0" t="0"></LT>
      </L>
      <T id="5" l="0" t="116.1" w="369.2" h="13" b="1">DOCMAPEND:00:DOCMAPEND  </T>
    </P>
  </Page>
  <Page id="2">
    <Header>
      <P storyId="2">
        <T id="6" l="0" t="2.9" w="47.2" h="13" b="1">Header </T>
      </P>
    </Header>
    <Footer>
      <P storyId="3">
        <T id="7" l="0" t="467.8" w="43.1" h="13" b="1">Footer </T>
      </P>
    </Footer>
    <P storyId="1">
      <T id="8" l="0" t="18.9" w="304.1" h="13">You can easily change the
formatting of selected text in </T>
      <T id="9" l="0" t="37.9" w="321.2" h="13">the document text by choosing a
look for the selected text </T>
      <T id="10" l="0" t="57" w="303.1" h="13">from the Quick Styles gallery on the
Home tab. You can </T>
      <T id="11" l="0" t="75" w="321.1" h="13">also format text directly by using
the other controls on the </T>
      <T id="12" l="0" t="94" w="309.1" h="13">Home tab. Most controls offer a
choice of using the look </T>
      <T id="13" l="0" t="113.1" w="318" h="13">from the current theme or using a
format specified.</T>
      <T id="14" l="0" t="129.2" w="58.2" h="16.9" b="1">directly.1 </T>
    </P>
    <P storyId="1" type="Heading">
      <T id="15" l="0" t="181.4" w="76.2" h="16" b="1">Header 1 </T>
    </P>
    <P storyId="1">
      <L l="-3" t="200.9" w="73.3" h="32.1" s="0" n="13">
        <LT p="2" l="-3" t="160.6"></LT>
      </L>
      <T id="16" l="0" t="205.3" w="76.2" h="13" b="1">Internal Link </T>
    </P>
    <P storyId="1">
      <L l="-3" t="232.9" w="72.7" h="32.1" s="0"
n="13">http://www.contoso.com/</L>
      <T id="17" l="0" t="237.3" w="77.2" h="13" b="1">External link </T>
    </P>
    <Footer>
      <P storyId="4">
        <T id="18" l="0" t="434.8" w="200.6" h="13" b="1">
</T>
      </P>
    </Footer>
    <Footnote>
      <P storyId="1">
        <T id="19" l="0" t="446.9" w="167.5" h="16.9" b="1">1 This is an example
footnote. </T>
      </P>
    </Footnote>
  </Page>
  <Page id="3">
    <Header>
      <P storyId="2">
        <T id="20" l="0" t="2.9" w="47.2" h="13" b="1">Header </T>
      </P>
    </Header>
    <Footer>
      <P storyId="3">
        <T id="21" l="0" t="467.8" w="43.1" h="13" b="1">Footer </T>
      </P>
    </Footer>
    <P storyId="1">
```

```
      <T id="22" l="0" t="256.5" w="127.4" h="16.9" b="1"> Text with an endnotei
</T>
    </P>
    <List i="0">
      <P storyId="1" id="0" type="ListBody">
        <T id="23" l="24.1" t="291.9" w="87.3" h="13.6" b="1">* Bullet one </T>
      </P>
      <P storyId="1" id="0" type="ListBody">
        <T id="24" l="24.1" t="312" w="87.3" h="13.6" b="1">* Bullet two </T>
      </P>
      <P storyId="1" id="0" type="ListBody">
        <T id="25" l="24.1" t="331" w="95.3" h="13.6" b="1">* Bullet three </T>
      </P>
    </List>
    <TOC>
      <P storyId="1" type="TOCI">
        <T id="26" l="0" t="380.8" w="74.2" h="16" b="1">Contents </T>
      </P>
      <P storyId="1" type="TOCI">
        <L l="-3" t="400.2" w="250.4" h="25.4" s="0" n="72">
          <LT p="2" l="-3" t="160.6"></LT>
        </L>
        <T id="27" l="0" t="403.7" w="252.8" h="13" b="1">Header 1
..............................................................1 </T>
      </P>
      <P storyId="1" type="TOCI">
        <T id="28" l="0" t="430.7" w="8" h="13" b="1"> </T>
      </P>
    </TOC>
    <Image type="Figure" l="1" t="16.3" w="319" h="271.7"/>
  </Page>
  <Page id="4">
    <Header>
      <P storyId="2">
        <T id="29" l="0" t="2.9" w="47.2" h="13" b="1">Header </T>
      </P>
    </Header>
    <Footer>
      <P storyId="3">
        <T id="30" l="0" t="467.8" w="43.1" h="13" b="1">Footer </T>
      </P>
    </Footer>
    <Table>
      <TR>
        <TD>
          <P storyId="1">
            <T id="31" l="0" t="18.9" w="67.2" h="13" b="1">Header cell </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="32" l="81.3" t="18.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="33" l="161.5" t="18.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="34" l="240.7" t="18.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
      </TR>
      <TR>
        <TD>
          <P storyId="1">
            <T id="35" l="0" t="35.9" w="53.2" h="13" b="1">Data cell </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="36" l="81.3" t="35.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="37" l="161.5" t="35.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="38" l="240.7" t="35.9" w="6" h="13" b="1"> </T>
          </P>
        </TD>
      </TR>
      <TR>
        <TD>
          <P storyId="1">
            <T id="39" l="0" t="52" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="40" l="81.3" t="52" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="41" l="161.5" t="52" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="42" l="240.7" t="52" w="6" h="13" b="1"> </T>
          </P>
        </TD>
      </TR>
      <TR>
        <TD>
          <P storyId="1">
            <T id="43" l="0" t="68" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="44" l="81.3" t="68" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
```

```
            <T id="45" l="161.5" t="68" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="46" l="240.7" t="68" w="6" h="13" b="1"> </T>
          </P>
        </TD>
      </TR>
      <TR>
        <TD>
          <P storyId="1">
            <T id="47" l="0" t="84" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="48" l="81.3" t="84" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="49" l="161.5" t="84" w="6" h="13" b="1"> </T>
          </P>
        </TD>
        <TD>
          <P storyId="1">
            <T id="50" l="240.7" t="84" w="6" h="13" b="1"> </T>
          </P>
        </TD>
      </TR>
    </Table>
    <P storyId="1">
      <T id="51" l="0" t="102.1" w="8" h="13" b="1"> </T>
    </P>
    <P storyId="1">
      <T id="52" l="0" t="134.1" w="8" h="13" b="1"> </T>
    </P>
    <Footer>
      <P storyId="4">
        <T id="53" l="0" t="165.2" w="200.6" h="13" b="1">
</T>
      </P>
    </Footer>
    <Endnote>
      <P storyId="1">
        <T id="54" l="0" t="177.3" w="162.5" h="16.9" b="1">i This is an example
endnote. </T>
      </P>
    </Endnote>
    <TextBox>
      <P storyId="5">
        <T id="55" l="10" t="172.2" w="107.4" h="13">[Type a quote from </T>
        <T id="56" l="10" t="191.3" w="94.3" h="13">the document or </T>
        <T id="57" l="10" t="210.3" w="105.4" h="13">the summary of an </T>
        <T id="58" l="10" t="228.3" w="96.3" h="13">interesting point. </T>
        <T id="59" l="10" t="247.4" w="94.4" h="13">You can position </T>
        <T id="60" l="10" t="266.4" w="68.2" h="13">the text box </T>
        <T id="61" l="10" t="284.4" w="91.3" h="13">anywhere in the </T>
        <T id="62" l="10" t="303.5" w="106.4" h="13">document. Use the </T>
        <T id="63" l="10" t="322.5" w="102.4" h="13">Text Box Tools tab </T>
```

```
           <T id="64" l="10" t="341.6" w="78.3" h="13">to change the </T>
           <T id="65" l="10" t="359.6" w="96.3" h="13">formatting of the </T>
           <T id="66" l="10" t="378.6" w="83.3" h="13">pull quote text </T>
           <T id="67" l="10" t="395.7" w="35.1" h="13" b="1">box.] </T>
         </P>
       </TextBox>
     </Page>
   </Pages>
```

5 Security

5.1 Security Considerations for Implementers

Secure Sockets Layer (SSL) is required to securely implement this protocol.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

None.

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SharePoint Foundation 2010
- Microsoft Word Mobile 2010
- Microsoft Word Online

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 1.3: Word Online supports only the .docx, .docm, .dotx, and .dotm file formats.

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