[MS-OXWSCONT]:

Contacts Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	2.0.0	Major	Updated and revised the technical content.
5/5/2010	2.0.1	Editorial	Revised and edited the technical content.
8/4/2010	3.0	Major	Significantly changed the technical content.
11/3/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	None	No changes to the meaning, language, and formatting of the technical content.
8/5/2011	3.1	Minor	Clarified the meaning of the technical content.
10/7/2011	3.1	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	4.0	Major	Significantly changed the technical content.
4/27/2012	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.1	Minor	Clarified the meaning of the technical content.
10/8/2012	4.2	Minor	Clarified the meaning of the technical content.
2/11/2013	5.0	Major	Significantly changed the technical content.
7/26/2013	6.0	Major	Significantly changed the technical content.
11/18/2013	7.0	Major	Significantly changed the technical content.
2/10/2014	7.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	7.1	Minor	Clarified the meaning of the technical content.
7/31/2014	8.0	Major	Significantly changed the technical content.
10/30/2014	8.1	Minor	Clarified the meaning of the technical content.
5/26/2015	9.0	Major	Significantly changed the technical content.
9/14/2015	10.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the Contacts Web Service protocol.

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 [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

contact: (1) A presence entity (presentity) whose presence information can be tracked.

(2) A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

email address: A string that identifies a user and enables the user to receive Internet messages.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

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].

mailbox: A **message store** that contains email, calendar items, and other Message objects for a single recipient.

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

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

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [RFC5751].

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

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

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

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

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 for more information.

SOAP message: An **XML** document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

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 server: A server computer that hosts websites and responds to requests from applications.

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.

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

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

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [XML1.0].

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

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

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

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

1.2 References

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

1.2.1 Normative References

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

[MS-OXWSCDATA] Microsoft Corporation, "Common Web Service Data Types".

[MS-OXWSCORE] Microsoft Corporation, "Core Items Web Service Protocol".

[MS-OXWSDLIST] Microsoft Corporation, "Distribution List Creation and Usage Web Service Protocol".

[MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol".

[MS-OXWSRSLNM] Microsoft Corporation, "Resolve Recipient Names Web Service Protocol".

[MS-OXWSXPROP] Microsoft Corporation, "Extended Properties Structure".

[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

[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", May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[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

[WSIBASIC] Ballinger, K., Ehnebuske, D., Gudgin, M., et al., Eds., "Basic Profile Version 1.0", Final Material, April 2004, http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[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., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "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

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol".

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

[MS-OXWSADISC] Microsoft Corporation, "Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol".

1.3   Overview

The Contacts Web Service protocol provides the messages needed to create, get, update, delete, move, and copy **contact (2)** items on the server.

1.4   Relationship to Other Protocols

A client that implements this protocol can use the Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], to identify the target **endpoint** to use for each operation.

This protocol uses **SOAP**, as described in [SOAP1.1], to specify the structure information that is exchanged between the client and the server. This protocol uses the **XML schema**, as described in [XMLSCHEMA1] and [XMLSCHEMA2], to describe the message content that is sent to and from the server.

This protocol uses SOAP over **HTTP**, as described in [RFC2616], and SOAP over **HTTPS**, as described in [RFC2818], as shown in the following layering diagram.
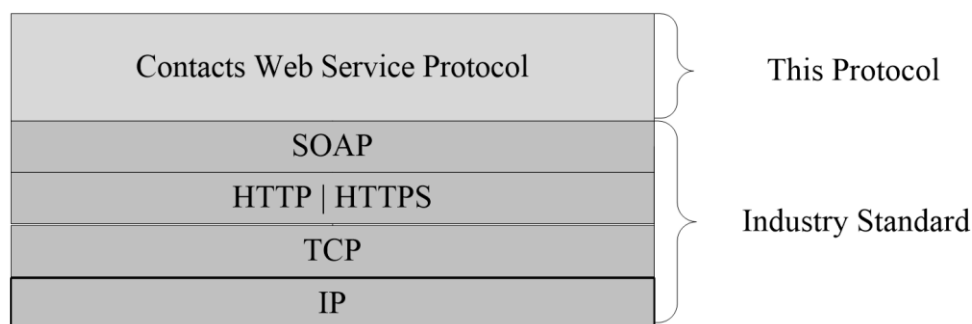
Figure 1: This protocol in relation to other protocols

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5   Prerequisites/Preconditions

The endpoint **URL** that is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], or known by the protocol client, is required to form the HTTP request to the **web server** that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP web requests that target this protocol..

1.6   Applicability Statement

This protocol is applicable to client programs that create, update, or manage contact (2) items in the server **message store**.

1.7   Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1. For more information, see section 2.1.

- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.

- **Security and Authentication Methods:** This protocol relies on the web server that is hosting it to perform authentication.

- **Localization:** This protocol uses the **MailboxCulture** part, as described in [MS-OXWSCORE] section 3.1.4.1.1.1, to specify the culture of a **mailbox**, and elements that are of the **xs:dateTime** type, as described in section 2.2.4.3.

- **Capability Negotiation:** None.

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 provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, or **present**.

2.1 Transport

The SOAP version supported is SOAP 1.1, as specified in [SOAP1.1].

The protocol MUST support SOAP over HTTP, as specified in [RFC2616]. The protocol SHOULD use secure communications by means of HTTPS, as specified in [RFC2818].

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 defined in [XMLSCHEMA1] and [XMLSCHEMA2], and WSDL, as defined in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA2]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	
m	http://schemas.microsoft.com/exchange/services/2006/messages	
wsi	http://ws-i.org/schemas/conformanceClaim/	[WSIBASIC]

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 that are defined by this specification. XML schema complex type definitions that are specific to a particular operation are defined with the operation.

Complex type	Description
t:AbchPersonItemType (section 2.2.4.1)	Specifies a person.
t:ArrayOfBinaryType (section 2.2.4.2)	Specifies a collection of certificates for a contact (2).
t:ContactItemType (section 2.2.4.3)	Represents a server contact (2) item.

2.2.4.1 t:AbchPersonItemType Complex Type

The **AbchPersonItemType** complex type specifies a person.<1> This type specifies the properties needed for a contact to use consumer accounts.

```
<xs:complexType name="AbchPersonItemType">
  <xs:complexContent>
    <xs:extension base="t:ItemType">
      <xs:sequence>
        <xs:element name="AntiLinkInfo" type="xs:string" minOccurs="0"/>
        <xs:element name="PersonId" type="t:GuidType" minOccurs="0"/>
        <xs:element name="ContactHandles" type="t:ArrayOfAbchPersonContactHandlesType"
minOccurs="0"/>
        <xs:element name="ContactCategories" type="t:ArrayOfStringsType" minOccurs="0"/>
        <xs:element name="RelevanceOrder1" type="xs:string" minOccurs="0"/>
        <xs:element name="RelevanceOrder2" type="xs:string" minOccurs="0"/>
        <xs:element name="TrustLevel" type="xs:int" minOccurs="0"/>
        <xs:element name="FavoriteOrder" type="xs:int" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table describes the child elements of the **AbchPersonItemType** complex type.

Element	Type	Description
AntiLinkInfo	**xs:string** ([XMLSCHEMA2])	An id of a set of people who should not be linked together automatically.
PersonId	**t:GuidType** ([MS-OXWSXPROP] section 2.1.7)	The id of this person.
ContactHandles	**t:ArrayOfAbchPersonContactHandlesType** (section 3.1.4.1.1.4)	A list of the handles to contacts represented by this person.
ContactCategories	**t:ArrayOfStringsType** ([MS-OXWSCDATA] section 2.2.4.13)	The categories of groups that this person belongs to.
RelevanceOrder1	**xs:string**	For internal use only.
RelevanceOrder2	**xs:string**	For internal use only.

Element	Type	Description
TrustLevel	**xs:int** ([XMLSCHEMA2])	For internal use only.
FavoriteOrder	**xs:int**	Implementation-specific favorite order. If this value is 0, this person is not a favorite. Otherwise, a non-zero value means this person is a favorite.

2.2.4.2 t:ArrayOfBinaryType Complex Type

The **ArrayOfBinaryType** complex type specifies a collection of certificates for a contact (2).<2> This type is used by the **UserSMIMECertificate** element and the **MSExchangeCertificate** element of the **ContactItemType** complex type, as specified in section 2.2.4.3.

```
<xs:complexType name="ArrayOfBinaryType">
  <xs:sequence>
    <xs:element name="Base64Binary" type="xs:base64Binary" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

Child element

Element name	Type	Description
Base64Binary	**xs:base64Binary** [XMLSCHEMA2]	Specifies a single certificate for a contact (2). The value is encoded with **base64 encoding**.

2.2.4.3 t:ContactItemType Complex Type

The **ContactItemType** complex type represents a server contact (2) item. It is also used by the **ResolveNames** method ([MS-OXWSRSLNM] section 3.1.4.1), returning directory and store contacts (2) matching a search string. This type extends the **ItemType** complex type, as specified in [MS-OXWSCORE] section 2.2.4.24. This type is used by the **CreateItem** operation, as specified in section 3.1.4.6, and the **UpdateItem** operation, as specified in section 3.1.4.3.

```
<xs:complexType name="ContactItemType">
  <xs:complexContent>
    <xs:extension
      base="t:ItemType"
    >
      <xs:sequence>
        <xs:element name="FileAs"
          type="xs:string"
          minOccurs="0"
         />
        <xs:element name="FileAsMapping"
          type="t:FileAsMappingType"
          minOccurs="0"
         />
        <xs:element name="DisplayName"
```

```
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="GivenName"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="Initials"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="MiddleName"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="Nickname"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="CompleteName"
        type="t:CompleteNameType"
        minOccurs="0"
       />
      <xs:element name="CompanyName"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="EmailAddresses"
        type="t:EmailAddressDictionaryType"
        minOccurs="0"
       />
      <xs:element name="AbchEmailAddresses"
        type="t:AbchEmailAddressDictionaryType"
        minOccurs="0"
       />
      <xs:element name="PhysicalAddresses"
        type="t:PhysicalAddressDictionaryType"
        minOccurs="0"
       />
      <xs:element name="PhoneNumbers"
        type="t:PhoneNumberDictionaryType"
        minOccurs="0"
       />
      <xs:element name="AssistantName"
        type="xs:string"
        minOccurs="0"
       />
      <xs:element name="Birthday"
        type="xs:dateTime"
        minOccurs="0"
       />
      <xs:element name="BusinessHomePage"
        type="xs:anyURI"
        minOccurs="0"
       />
      <xs:element name="Children"
        type="t:ArrayOfStringsType"
        minOccurs="0"
       />
      <xs:element name="Companies"
        type="t:ArrayOfStringsType"
        minOccurs="0"
       />
      <xs:element name="ContactSource"
        type="t:ContactSourceType"
        minOccurs="0"
       />
      <xs:element name="Department"
        type="xs:string"
```

```
          minOccurs="0"
 />
 <xs:element name="Generation"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="ImAddresses"
   type="t:ImAddressDictionaryType"
   minOccurs="0"
 />
 <xs:element name="JobTitle"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="Manager"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="Mileage"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="OfficeLocation"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="PostalAddressIndex"
   type="t:PhysicalAddressIndexType"
   minOccurs="0"
 />
 <xs:element name="Profession"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="SpouseName"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="Surname"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="WeddingAnniversary"
   type="xs:dateTime"
   minOccurs="0"
 />
 <xs:element name="HasPicture"
   type="xs:boolean"
   minOccurs="0"
 />
 <xs:element name="PhoneticFullName"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="PhoneticFirstName"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="PhoneticLastName"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="Alias"
   type="xs:string"
   minOccurs="0"
 />
 <xs:element name="Notes"
   type="xs:string"
   minOccurs="0"
```

```
  />
  <xs:element name="Photo"
    type="xs:base64Binary"
    minOccurs="0"
  />
  <xs:element name="UserSMIMECertificate"
    type="t:ArrayOfBinaryType"
    minOccurs="0"
  />
  <xs:element name="MSExchangeCertificate"
    type="t:ArrayOfBinaryType"
    minOccurs="0"
  />
  <xs:element name="DirectoryId"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="ManagerMailbox"
    type="t:SingleRecipientType"
    minOccurs="0"
  />
  <xs:element name="DirectReports"
    type="t:ArrayOfRecipientsType"
    minOccurs="0"
  />
  <xs:element name="AccountName"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="IsAutoUpdateDisabled"
    type="xs:boolean"
    minOccurs="0"
  />
  <xs:element name="IsMessengerEnabled"
    type="xs:boolean"
    minOccurs="0"
  />
  <xs:element name="Comment"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="ContactShortId"
    type="xs:int"
    minOccurs="0"
  />
  <xs:element name="ContactType"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="Gender"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="IsHidden"
    type="xs:boolean"
    minOccurs="0"
  />
  <xs:element name="ObjectId"
    type="xs:string"
    minOccurs="0"
  />
  <xs:element name="PassportId"
    type="xs:long"
    minOccurs="0"
  />
  <xs:element name="PersonId"
    type="xs:int"
    minOccurs="0"
  />
```

```
<xs:element name="IsPrivate"
  type="xs:boolean"
  minOccurs="0"
/>
<xs:element name="SourceId"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="TrustLevel"
  type="xs:int"
  minOccurs="0"
/>
<xs:element name="CreatedBy"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="Urls"
  type="t:ContactUrlDictionaryType"
  minOccurs="0"
/>
<xs:element name="Cid"
  type="xs:long"
  minOccurs="0"
/>
<xs:element name="SkypeAuthCertificate"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="SkypeContext"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="SkypeId"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="SkypeRelationship"
  type="xs:string"
  minOccurs="0"
/>

<xs:element name="YomiNickname"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="XboxLiveTag"
  type="xs:string"
  minOccurs="0"
/>
<xs:element name="InviteFree"
  type="xs:boolean"
  minOccurs="0"
/>
<xs:element name="HidePresenceAndProfile"
  type="xs:boolean"
  minOccurs="0"
/>
<xs:element name="IsPendingOutbound"
  type="xs:boolean"
  minOccurs="0"
/>
<xs:element name="SupportGroupFeeds"
  type="xs:boolean"
  minOccurs="0"
/>
<xs:element name="UserTileHash"
  type="xs:string"
  minOccurs="0"
```

```
          />
          <xs:element name="UnifiedInbox"
            type="xs:boolean"
            minOccurs="0"
          />
          <xs:element name="Mris"
            type="t:ArrayOfStringsType"
            minOccurs="0"
          />
          <xs:element name="Mris"
            type="t:ArrayOfStringsType"
            minOccurs="0"
          />
          <xs:element name="Wlid"
            type="xs:string"
            minOccurs="0"
          />
          <xs:element name="AbchContactId"
            type="t:GuidType"
            minOccurs="0"
          />
          <xs:element name="NotInBirthdayCalendar"
            type="xs:boolean"
            minOccurs="0"
          />


    <

xs:element name="ShellContactType"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name ="ImMri"
          type="xs:int"
          minOccurs="0"
        />
        <xs:element name="PresenceTrustLevel"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="OtherMri"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="ProfileLastChanged"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="MobileImEnabled"
          type="xs:boolean"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:entension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements<3>

Element name	Type	Description
FileAs	**xs:string** [XMLSCHEMA2]	Represents how a contact (2) is filed in the Contacts folder.

Element name	Type	Description
FileAsMapping	**t:FileAsMappingType** (section 3.1.4.1.2.4)	Defines how to construct what is displayed for a contact (2).
DisplayName	**xs:string**	Contains the display name of a contact (2).
GivenName	**xs:string**	Contains the given name for a contact (2).
Initials	**xs:string**	Contains the initials for a contact (2).
MiddleName	**xs:string**	Represents the middle name of a contact (2).
Nickname	**xs:string**	Represents the nickname of a contact (2).
CompleteName	**t:CompleteNameType** (section 3.1.4.1.1.5)	Represents the complete name of a contact (2). This property is read-only for client.
CompanyName	**xs:string**	Contains the company name that is associated with a contact (2).
EmailAddresses	**t:EmailAddressDictionaryType** (section 3.1.4.1.1.11)	Contains email addresses that are associated with a contact (2). If the email address is invalid, server MUST return response code **ErrorInvalidContactEmailAddress** ([MS-OXWSCDATA] section 2.2.5.24).
AbchEmailAddresses	**t:AbchEmailAddressDictionaryType (**section 3.1.4.1.1.2**)**	Represents an email address entry.
PhysicalAddresses	**t:PhysicalAddressDictionaryType** (section 3.1.4.1.1.17)	Represents a collection of physical addresses that are associated with a contact (2).
PhoneNumbers	**t:PhoneNumberDictionaryType** (section 3.1.4.1.1.15)	Represents a collection of telephone numbers for a contact (2).
AssistantName	**xs:string**	Contains the name of the assistant for the contact (2).
Birthday	**xs:dateTime** [XMLSCHEMA2]	Represents the birthday of the contact (2).
BusinessHomePage	**xs:anyURI** [XMLSCHEMA2]	Contains the business home page **Uniform Resource Identifier (URI)** of a contact (2).
Children	**t:ArrayOfStringsType** ([MS-OXWSCDATA] section 2.2.4.13)	Contains the names of children for the contact (2).
Companies	**t:ArrayOfStringsType**	Contains the names of companies that are associated with a contact (2).
ContactSource	**t:ContactSourceType** (section 2.2.5.1)	Describes whether the contact (2) is located in the server message store or the directory service.
Department	**xs:string**	Contains the work department for the contact (2).
Generation	**xs:string**	Contains a generational abbreviation that follows the full name of a contact (2).
ImAddresses	**t:ImAddressDictionaryType** (section 3.1.4.1.1.13)	Contains instant messaging addresses for a contact (2).

Element name	Type	Description
JobTitle	**xs:string**	Contains the job title of a contact (2).
Manager	**xs:string**	Represents the manager of a contact (2).
Mileage	**xs:string**	Represents the mileage for a contact (2).
OfficeLocation	**xs:string**	Represents the office location of a contact (2).
PostalAddressIndex	**t:PhysicalAddressIndexType** (section 3.1.4.1.2.7)	Represents the index of one of the physical addresses, which is a contact's (2) mailing address.
Profession	**xs:string**	Represents the profession of a contact (2).
SpouseName	**xs:string**	Represents the name of the spouse/partner of a contact (2).
Surname	**xs:string**	Contains the surname of a contact (2).
WeddingAnniversary	**xs:dateTime**	Contains the wedding anniversary date of a contact (2).
HasPicture	**xs:boolean** [XMLSCHEMA2]	Represents that the contact (2) has a picture. This element is read-only for the client.<4>
PhoneticFullName	**xs:string**	Contains the full name of a contact (2), including the first and last name, spelled phonetically.
PhoneticFirstName	**xs:string**	Contains the first name of a contact (2), spelled phonetically.
PhoneticLastName	**xs:string**	Contains the last name of a contact (2), spelled phonetically.
Alias	**xs:string**	Contains the email alias of a contact.
Notes	**xs:string**	Contains supplementary contact (2) information.
Photo	**xs:base64Binary** [XMLSCHEMA2]	Contains a value that encodes the photo of a contact (2).
UserSMIMECertificate	**t:ArrayOfBinaryType** (section 2.2.4.2)	Contains a value that encodes a contacts (2) **S/MIME** certificate.
MSExchangeCertificate	**t:ArrayOfBinaryType**	Contains a value that encodes the server certificate of a contact (2).
DirectoryId	**xs:string**	Contains the directory identifier of a contact (2).
ManagerMailbox	**t:SingleRecipientType** ([MS-OXWSCDATA] section 2.2.4.71)	Contains **SMTP** information that identifies the mailbox of a contact's (2) manager.
DirectReports	**t:ArrayOfRecipientsType** ([MS-OXWSCDATA] section 2.2.4.11)	Contains SMTP information that identifies the mailboxes of a contact's (2) direct reports.
AccountName	**xs:string**	The account name. <5>

Element name	Type	Description
IsAutoUpdateDisabled	**xs:boolean**	Specifies whether auto-update is disabled for this contact.<6>
IsMessengerEnabled	**xs:boolean**	For internal use only.
Comment	**xs:string**	A comment.<7>
ContactShortId	**xs:int**	For internal use only.<8>
ContactType	**xs:string**	The type of the contact.<9>
Gender	**xs:string**	The gender of the contact.<10>
IsHidden	**xs:boolean**	For internal use only.<11>
ObjectId	**xs:string**	The handle of the contact.<12>
PassportId	**xs:long**	For internal use only.<13>
PersonId	**xs:int**	Specifies the id of the person.<14>
IsPrivate	**xs:boolean**	For internal use only.<15>
SourceId	**xs:string**	Specifies the source of the contact.<16>
TrustLevel	**xs:int**	For internal use only.<17>
CreatedBy	**xs:string**	For internal use only.<18>
Urls	**t:ContactUrlDictionaryType (**section 3.1.4.1.1.9**)**	Specifies the websites for the contact.<19>
Cid	**xs:long**	Specifies a unique handle to identify a user.<20>
SkypeAuthCertificate	**xs:string**	Specifies a token to connect to Skype.<21>
SkypeContext	**xs:string**	For internal use only.<22>
SkypeId	**xs:string**	The account name on Skype.<23>
SkypeRelationship	**xs:string**	For internal use only.<24>
YomiNickname	**xs:string**	Pronunciation guide for nickname field.<25>
XboxLiveTag	**xs:string**	Not used.<26>
InviteFree	**xs:boolean**	For internal use only.<27>
HidePresenceAndProfile	**xs:boolean**	For internal use only.<28>
IsPendingOutbound	**xs:boolean**	For internal use only.<29>
SupportGroupFeeds	**xs:boolean**	Not used.<30>
UserTileHash	**xs:string**	Not used.<31>
UnifiedInbox	**xs:boolean**	For internal use only.<32>
Mris	**t:ArrayOfStringsType**	For internal use only.<33>
Wlid	**xs:string**	For internal use only.<34>

Element name	Type	Description
AbchContactId	**t:GuidType**	For internal use only.<35>
NotInBirthdayCalendar	**xs:boolean**	Not used.<36>

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
t:ContactSourceType (section 2.2.5.1)	Specifies whether a contact (2) or **distribution list** is located in the server database or in the directory service.

2.2.5.1 t:ContactSourceType Simple Type

The **ContactSourceType** specifies whether a contact (2) or distribution list is located in the server database or in the directory service.

```
<xs:simpleType name="ContactSourceType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="ActiveDirectory"
     />
    <xs:enumeration
      value="Store"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **ContactSourceType** simple type.

Value	Meaning
ActiveDirectory	Specifies that the contact (2) or distribution list is located in the directory service.
Store	Specifies that the contact (2) or distribution list is located in the server database.

This is applicable to the Contacts Web Service protocol and to the Distribution List Creation and Usage Web Service protocol [MS-OXWSDLIST].

It is also used by the **ResolveNames** method ([MS-OXWSRSLNM] section 3.1.4.1), returning directory and store contacts (2) matching a search string.

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.

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.

3.1   ExchangeServicePortType Server Details

This protocol defines a single port type.

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Message Processing Events and Sequencing Rules

This protocol includes the operations listed in the following table.

Operation	Description
CopyItem (section 3.1.4.5)	Defines a request to copy an item in a mailbox in the server.
CreateItem (section 3.1.4.6)	Defines a request to create an item in the server.
DeleteItem (section 3.1.4.2)	Defines a request to delete an item from a mailbox in the server.
GetItem (section 3.1.4.1)	Defines a request to get an item from a mailbox in the server.
MoveItem (section 3.1.4.4)	Defines a request to move an item in the server.
UpdateItem (section 3.1.4.3)	Defines a request to update an item in a mailbox.

3.1.4.1 GetItem

This protocol uses the **GetItem** operation specified in [MS-OXWSCORE] section 3.1.4.4 to get contact (2) item elements.

```
<wsdl:operation name="GetItem">
        <wsdl:input message="tns:GetItemSoapIn" />
        <wsdl:output message="tns:GetItemSoapOut" />
    </wsdl:operation>
```

Request

Message Format	Description
tns:GetItemSoapIn ([MS-OXWSCORE] section 3.1.4.4.1.1)	Specifies the **SOAP message** that defines the contact (2) item to get. The **Items** ([MS-OXWSCDATA] section 2.2.4.48) child element of the **GetItem** ([MS-OXWSCORE] section 3.1.4.4.2.1) child element that specifies the **XML** request MUST contain the following elements: **t:ItemResponseShapeType** ([MS-OXWSCDATA] section 2.2.4.44), **t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25). All other elements MUST be empty.

Response

Message Format	Description
tns:GetItemSoapOut ([MS-OXWSCORE] section 3.1.4.4.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.1.1 Complex Types

The following XML schema complex type definitions are specific to this operation.

3.1.4.1.1.1   t:AbchEmailAddressDictionaryEntryType Complex Type

The **AbchEmailAddressDictionaryEntryType** complex type specifies an email address entry.<37>

```
<xs:complexType name="AbchEmailAddressDictionaryEntryType">
  <xs:sequence>
    <xs:element name="Type" type="t:AbchEmailAddressTypeType" minOccurs="1"/>
    <xs:element name="Address" type="xs:string" minOccurs="1"/>
    <xs:element name="IsMessengerEnabled" type="xs:boolean" minOccurs="0"/>
    <xs:element name="Capabilities" type="xs:long" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The following table describes the child elements of the **AbchEmailAddressDictionaryEntryType** complex type.

Element	Type	Description
Type	**t:AbchEmailAddressTypeType** (section 3.1.4.1.2.1)	Specifies the email type.
Address	**xs:string** ([XMLSCHEMA2])	Specifies the email address.
IsMessengerEnabled	**xs:boolean** ([XMLSCHEMA2])	Specifies whether this address is enambeld for Microsoft Skype for Business 2016.
Capabilities	**xs:long** ([XMLSCHEMA2])	For internal use only.

3.1.4.1.1.2   t:AbchEmailAddressDictionaryType Complex Type

The **AbchEmailAddressDictionaryType** complex type specifies an email address entry.<38>

```
<xs:complexType name="AbchEmailAddressDictionaryType">
  <xs:sequence>
```

```
      <xs:element name="Email" type="t:AbchEmailAddressDictionaryEntryType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

The following table describes the child element of the **AbchEmailAddressDictionaryType** complex type.

Element	Type	Description
Email	**t:AbchEmailAddressDictionaryEntryType** (section 3.1.4.1.1.2)	An email address entry.

3.1.4.1.1.3   t:AbchPersonContactHandle Complex Type

The **AbchPersonContactHandle** complex type specifies the link between a person item and contact.<39>

```
    <xs:complexType name="AbchPersonContactHandle">
      <xs:sequence>
        <xs:element name="SourceId" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ObjectId" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="AccountName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
```

The following table describes the child elements of the **AbchPersonContactHandle** complex type.

Element	Type	Description
SourceId	**xs:string** ([XMLSCHEMA2])	The source of the contact.
ObjectId	**xs:string**	Specifies the unique id within the source.
AccountName	**xs:string**	Differentiator if more than 1 entry.

3.1.4.1.1.4   t:ArrayOfAbchPersonContactHandlesType Complex Type

The **ArrayOfAbchPersonContactHandlesType** complex type specifies a contact handle.<40>

```
    <xs:complexType name="ArrayOfAbchPersonContactHandlesType">
      <xs:sequence>
        <xs:element name="ContactHandle" type="t:AbchPersonContactHandle" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
```

The following table lists the child element of the **ArrayOfAbchPersonContactHandlesType** complex type.

Element	Type	Description
ContactHandle	**t:AbchPersonContactHandle** (section 3.1.4.1.1.3)	The contact handle.

3.1.4.1.1.5 t:CompleteNameType Complex Type

The **CompleteNameType** complex type represents the complete name of a contact (2).

```
<xs:complexType name="CompleteNameType">
  <xs:sequence>
    <xs:element name="Title"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="FirstName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="MiddleName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="LastName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Suffix"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Initials"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="FullName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Nickname"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="YomiFirstName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="YomiLastName"
      type="xs:string"
      minOccurs="0"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element name	Type	Description
Title	**xs:string** [XMLSCHEMA2]	Contains the title of a contact (2).

Element name	Type	Description
FirstName	**xs:string**	Contains the first name of a contact (2). This is the same as GivenName.
MiddleName	**xs:string**	Contains the middle name of a contact (2).
LastName	**xs:string**	Contains the last name of a contact (2). This is the same as the Surname.
Suffix	**xs:string**	Contains a suffix to a contact's (2) name. This is the same as the Generation property.
Initials	**xs:string**	Contains the initials of a contact (2).
FullName	**xs:string**	Contains the full name of a contact (2). This is the same as the DisplayName.
Nickname	**xs:string**	Contains the nickname of a contact (2).
YomiFirstName	**xs:string**	Contains the name used in Japan for the searchable or phonetic spelling of a Japanese first name.
YomiLastName	**xs:string**	Contains the name used in Japan for the searchable or phonetic spelling of a Japanese last name.

3.1.4.1.1.6 t:ContactsFolderType Complex Type

The **ContactsFolderType** complex type represents a Contacts folder in a mailbox.

```
<xs:complexType name="ContactsFolderType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseFolderType"
    >
      <xs:sequence>
        <xs:element name="SharingEffectiveRights"
          type="t:PermissionReadAccessType"
          minOccurs="0"
        />
        <xs:element name="PermissionSet"
          type="t:PermissionSetType"
          minOccurs="0"
        />
        <xs:element name="SourceId"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="AccountName"
          type="xs:string"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element name	Type	Description
SharingEffectiveRights	**t:PermissionReadAccessType** ([MS-OXWSFOLD] section 2.2.5.4)	Specifies whether a user has **permission** to read items in a folder. This element is read-only for the client.<41>
PermissionSet	**t:PermissionSetType** ([MS-OXWSFOLD] section 2.2.4.14)	Contains all the permissions that are configured for a folder.
SourceId	**xs:string** [XMLSCHEMA2]	Specifies the source Id.<42>
AccountName	**xs:string**	Specifies the account name.<43>

3.1.4.1.1.7   t:ContactsViewType Complex Type

The **ContactsViewType** complex type represents the settings that are used to return contact (2) items based on their alphabetical display names.

```
<xs:complexType name="ContactsViewType">
  <xs:complexContent>
    <xs:extension
      base="t:BasePagingType"
    >
      <xs:attribute name="InitialName"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="FinalName"
        type="xs:string"
        use="optional"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Attributes

Attribute name	Type	Description
InitialName	**xs:string** [XMLSCHEMA2]	Contains the first name in a contacts (2) list to return in a response.
FinalName	**xs:string**	Contains the last name in a contacts list to return in a response.

3.1.4.1.1.8   t:ContactUrlDictionaryEntryType Complex Type

The **ContactUrlDictionaryEntryType** complex type represents a URL that is associated with a contact.<44>

```
<xs:complexType name="ContactUrlDictionaryEntryType">
  <xs:sequence>
    <xs:element name="Type" type="t:ContactUrlKeyType" minOccurs="1"/>
    <xs:element name="Name" type="xs:string" minOccurs="0"/>
    <xs:element name="Address" type="xs:string" minOccurs="0"/>
  </xs:sequence>
```

```
     </xs:complexType>
```

The following table lists the child elements of the **ContactUrlDictionaryEntryType** complex type.

Element	Type	Description
Type	**t:ContactUrlKeyType (**section 3.1.4.1.2.2**)**	The Url type.
Name	**xs:string (**[XMLSCHEMA2])	What the url is used for.
Address	**xs:string**	The Url.

3.1.4.1.1.9   t:ContactUrlDictionaryType Complex Type

The **ContactUrlDictionaryType** complex type specifies a Url associated with a contact <45>

```
<xs:complexType name="ContactUrlDictionaryType">
  <xs:sequence>
    <xs:element name="Url" type="t:ContactUrlDictionaryEntryType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

The following table describes the child element of the **ContactUrlDictionaryType** complex type.

Element	Type	Description
Url	**t:ContactUrlDictionaryEntryType (**section 3.1.4.1.1.8**)**	A Url associated with a contact.

3.1.4.1.1.10 t:EmailAddressDictionaryEntryType Complex Type

The **EmailAddressDictionaryEntryType** complex type represents an e-mail address that is associated with a contact (2).

```
<xs:complexType name="EmailAddressDictionaryEntryType">
  <xs:simpleContent>
    <xs:extension
      base="xs:string"
    >
      <xs:attribute name="Key"
        type="t:EmailAddressKeyType"
        use="required"
       />
      <xs:attribute name="Name"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="RoutingType"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="MailboxType"
        type="t:MailboxTypeType"
        use="optional"
```

```
      />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Attributes

Name	Type	Description
Key	**t:EmailAddressKeyType** (section 3.1.4.1.2.3)	Contains a value that identifies an e-mail address that is associated with a contact (2). If the **Key** attribute is set to an invalid value, server MUST return response code **ErrorInvalidContactEmailIndex** ([MS-OXWSCDATA] section 2.2.5.24).
Name<46>	**xs:string** [XMLSCHEMA2]	Contains the display name associated with an e-mail address of the contact (2).
RoutingType<47>	**xs:string**	Contains the routing type associated with an e-mail address of the contact (2).
MailboxType<48>	**t:MailboxTypeType** ([MS-OXWSCDATA] section 2.2.5.18)	Contains the type of mailbox that is represented by the e-mail address of the contact (2).

3.1.4.1.1.11 t:EmailAddressDictionaryType Complex Type

The **EmailAddressDictionaryType** complex type contains e-mail addresses.

```
<xs:complexType name="EmailAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:EmailAddressDictionaryEntryType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element name	Type	Description
Entry	**t:EmailAddressDictionaryEntryType** (section 3.1.4.1.1.10)	Represents an e-mail address that is associated with a contact (2).

3.1.4.1.1.12 t:ImAddressDictionaryEntryType Complex Type

The **ImAddressDictionaryEntryType** complex type represents a collection of instant messaging addresses for a contact (2).

```
<xs:complexType name="ImAddressDictionaryEntryType">
  <xs:simpleContent>
```

```
<xs:extension
  base="xs:string"
>
  <xs:attribute name="key"
    type="t:ImAddressKeyType"
    use="required"
  />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
```

Attributes

Attribute name	Type	Description
key	**t:ImAddressKeyType** (section 3.1.4.1.2.5)	Represents the instant messaging addresses for a contact (2).

3.1.4.1.1.13 t:ImAddressDictionaryType Complex Type

The **ImAddressDictionaryType** complex type contains instant messaging addresses for a contact (2).

```
<xs:complexType name="ImAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:ImAddressDictionaryEntryType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element name	Type	Description
Entry	**t:ImAddressDictionaryEntryType** (section 3.1.4.1.1.14)	Represents a collection of instant messaging addresses for a contact (2).

3.1.4.1.1.14 t:PhoneNumberDictionaryEntryType Complex Type

The **PhoneNumberDictionaryEntryType** complex type contains a telephone number for a contact (2).

```
<xs:complexType name="PhoneNumberDictionaryEntryType">
  <xs:simpleContent>
    <xs:extension
      base="xs:string"
    >
      <xs:attribute name="Key"
        type="t:PhoneNumberKeyType"
```

```
      use="required"
    />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
```

Attributes

Attribute name	Type	Description
Key	**t:PhoneNumberKeyType** (section 3.1.4.1.2.6)	Represents types of telephone numbers for a contact (2).

3.1.4.1.1.15 t:PhoneNumberDictionaryType Complex Type

The **PhoneNumberDictionaryType** complex type represents telephone numbers for a contact (2).

```
<xs:complexType name="PhoneNumberDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:PhoneNumberDictionaryEntryType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element name	Type	Description
Entry	**t:PhoneNumberDictionaryEntryType** (section 3.1.4.1.1.14)	Contains a telephone number for a contact (2).

3.1.4.1.1.16 t:PhysicalAddressDictionaryEntryType Complex Type

The **PhysicalAddressDictionaryEntryType** complex type contains information that defines a physical address, such as a street address.

```
<xs:complexType name="PhysicalAddressDictionaryEntryType">
  <xs:sequence>
    <xs:element name="Street"
      type="xs:string"
      minOccurs="0"
    />
    <xs:element name="City"
      type="xs:string"
      minOccurs="0"
    />
    <xs:element name="State"
      type="xs:string"
      minOccurs="0"
    />
```

```
    <xs:element name="CountryOrRegion"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="PostalCode"
      type="xs:string"
      minOccurs="0"
     />
  </xs:sequence>
  <xs:attribute name="Key"
    type="t:PhysicalAddressKeyType"
    use="required"
   />
</xs:complexType>
```

Child Elements

Element name	Type	Description
Street	**xs:string** [XMLSCHEMA2]	Contains the street address for a contact (2) item.
City	**xs:string**	Contains the city name for a contact (2) item.
State	**xs:string**	Contains the state for a contact (2) item.
CountryOrRegion	**xs:string**	Contains the country or region for a contact (2) item.
PostalCode	**xs:string**	Contains the postal code for a contact (2) item.

Attributes

Attribute name	Type	Description
Key	**t:PhysicalAddressKeyType** (section 3.1.4.1.2.8)	Identifies the types of physical addresses for a contact (2).

3.1.4.1.1.17 t:PhysicalAddressDictionaryType Complex Type

The **PhysicalAddressDictionaryType** complex type contains physical addresses that are associated with a contact (2).

```
<xs:complexType name="PhysicalAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:PhysicalAddressDictionaryEntryType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element name	Type	Description
Entry	**t:PhysicalAddressDictionaryEntryType** (section 3.1.4.1.1.16)	Contains information that defines a physical address, such as a street address.

3.1.4.1.2 Simple Types

The following XML schema simple type definitions are specific to this operation.

3.1.4.1.2.1 t:AbchEmailAddressTypeType Simple Type

The **AbchEmailAddressTypeType** simple type specifies an email address type.<49>

```
<xs:simpleType name="AbchEmailAddressTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Personal"/>
    <xs:enumeration value="Business"/>
    <xs:enumeration value="Other"/>
    <xs:enumeration value="Passport"/>
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values of the **AbchEmailAddressTypeType** simple type.

Value	Meaning
Personal	Personal email.
Business	Business email.
Other	Some other type.
Passport	Microsoft email account.

3.1.4.1.2.2 t:ContactUrlKeyType Simple Type

The **ContactUrlKeyType** simple type represents types of URLs for a contact.<50>

```
<xs:simpleType name="ContactUrlKeyType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Personal"/>
    <xs:enumeration value="Business"/>
    <xs:enumeration value="Attachment"/>
    <xs:enumeration value="EbcDisplayDefinition"/>
    <xs:enumeration value="EbcFinalImage"/>
    <xs:enumeration value="EbcLogo"/>
    <xs:enumeration value="Feed"/>
    <xs:enumeration value="Image"/>
    <xs:enumeration value="InternalMarker"/>
    <xs:enumeration value="Other"/>
  </xs:restriction>
</xs:simpleType>
```

The following table lists the values of the **ContactUrlKeyType** simple type.

Value	Meaning
Personal	A personal Url.
Business	A business Url.
Attachment	An attachment Url.
EbcDisplayDefinition	Not used
EbcFinalImage	Not used
EbcLogo	Not used
Feed	Url for a news feed.
Image	Url to the contact image.
InternalMarker	For internal use only.
Other	For anything else.

3.1.4.1.2.3   t:EmailAddressKeyType Simple Type

The **EmailAddressKeyType** simple type represents a way to identify a single e-mail address within the e-mail address collection for a contact (2).

```
<xs:simpleType name="EmailAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="EmailAddress1"
     />
    <xs:enumeration
      value="EmailAddress2"
     />
    <xs:enumeration
      value="EmailAddress3"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **EmailAddressKeyType** simple type:

Value	Meaning
EmailAddress1	Identifies the first e-mail address for the contact (2).
EmailAddress2	Identifies the second e-mail address for the contact (2).
EmailAddress3	Identifies the third e-mail address for the contact (2).

3.1.4.1.2.4   t:FileAsMappingType Simple Type

The **FileAsMappingType** simple type defines how to construct what is displayed for a contact (2) in the **FileAs** property.

```
<xs:simpleType name="FileAsMappingType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="LastCommaFirst"
     />
    <xs:enumeration
      value="FirstSpaceLast"
     />
    <xs:enumeration
      value="Company"
     />
    <xs:enumeration
      value="LastCommaFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastFirst"
     />
    <xs:enumeration
      value="LastFirst"
     />
    <xs:enumeration
      value="LastFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastCommaFirst"
     />
    <xs:enumeration
      value="LastFirstSuffix"
     />
    <xs:enumeration
      value="LastSpaceFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastSpaceFirst"
     />
    <xs:enumeration
      value="LastSpaceFirst"
     />
    <xs:enumeration
      value="DisplayName"
     />
    <xs:enumeration
      value="FirstName"
     />
    <xs:enumeration
      value="LastFirstMiddleSuffix"
     />
    <xs:enumeration
      value="LastName"
     />
    <xs:enumeration
      value="Empty"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **FileAsMappingType** simple type:

Value	Meaning
None	Indicates that the **FileAs** value is not constructed from properties of other contacts (2), but is represented by a string, saved "as is".
LastCommaFirst	Indicates that the contact (2) is displayed as the last name followed by a comma, a space, the first name, a space, and the middle name.
FirstSpaceLast	Indicates that the contact (2) is displayed as the first name followed by a space, the middle name, a space, the last name, a space, and the suffix for the contact (2).
Company	Indicates that the company name is displayed.
LastCommaFirstCompany	Indicates that the contact (2) is displayed as the last name, a comma, a space, the first name, a space, the middle name, a space, a left parenthesis, the company name, and a right parenthesis.
CompanyLastFirst	Indicates that the contact (2) is displayed as the company name, a space, a left parenthesis, the last name, the first name, a space, the middle name, and a right parenthesis.
LastFirst	Indicates that the contact (2) is displayed as the last name followed by the first name, a space, and the middle name.
LastFirstCompany	Indicates that the contact (2) is displayed as the last name, the first name, a space, the middle name, a space, a left parenthesis, the company name, and a right parenthesis.
CompanyLastCommaFirst	Indicates that the contact (2) is displayed as the company name, a space, a left parenthesis, the last name, a comma, a space, and the first name, a space, the middle name, and a right parenthesis.
LastFirstSuffix	Indicates that the contact (2) is displayed as the last name, the first name, a space, and the suffix for the contact.
LastSpaceFirstCompany	Indicates that the contact (2) is displayed as the last name, a space, the first name, a space, the middle name, a space, a left parenthesis, the company name, and a right parenthesis.
CompanyLastSpaceFirst	Indicates that the contact (2) is displayed as the company name, a space, a left parenthesis, the last name, a space, the first name, a space, the middle name, and a right parenthesis.
LastSpaceFirst	Indicates that the contact (2) is displayed as the last name, followed by a space, the first name, a space, and the middle name.
DisplayName[51]	Indicates that the contact (2) is displayed as the display name.
FirstName[52]	Indicates that the contact (2) is displayed as the first name.
LastFirstMiddleSuffix[53]	Indicates that the contact (2) is displayed as the last name, the first name, the middle name, and the suffix for the contact.
LastName[54]	Indicates that the contact (2) is displayed as the last name.
Empty[55]	Indicates that the contact (2) is displayed as empty.

3.1.4.1.2.5   t:ImAddressKeyType Simple Type

The **ImAddressKeyType** enumeration represents the instant messaging addresses for a contact (2).

```
<xs:simpleType name="ImAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="ImAddress1"
     />
    <xs:enumeration
      value="ImAddress2"
     />
    <xs:enumeration
      value="ImAddress3"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **ImAddressKeyType** simple type:

Value	Meaning
ImAddress1	Identifies the first instant messaging address for the user.
ImAddress2	Identifies the second instant messaging address for the user.
ImAddress3	Identifies the third instant messaging address for the user.

3.1.4.1.2.6   t:PhoneNumberKeyType Simple Type

The **PhoneNumberKeyType** simple type represents types of telephone numbers for a contact (2).

```
<xs:simpleType name="PhoneNumberKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="AssistantPhone"
     />
    <xs:enumeration
      value="BusinessFax"
     />
    <xs:enumeration
      value="BusinessPhone"
     />
    <xs:enumeration
      value="BusinessPhone2"
     />
    <xs:enumeration
      value="Callback"
     />
    <xs:enumeration
      value="CarPhone"
     />
    <xs:enumeration
      value="CompanyMainPhone"
     />
    <xs:enumeration
      value="HomeFax"
     />
    <xs:enumeration
```

```
        value="HomePhone"
      />
    <xs:enumeration
      value="HomePhone2"
      />
    <xs:enumeration
      value="Isdn"
      />
    <xs:enumeration
      value="MobilePhone"
      />
    <xs:enumeration
      value="OtherFax"
      />
    <xs:enumeration
      value="OtherTelephone"
      />
    <xs:enumeration
      value="Pager"
      />
    <xs:enumeration
      value="PrimaryPhone"
      />
    <xs:enumeration
      value="RadioPhone"
      />
    <xs:enumeration
      value="Telex"
      />
    <xs:enumeration
      value="TtyTddPhone"
      />
    <xs:enumeration
      value="BusinessMobile"/>
    <xs:enumeration
      value="IPPhone"/>
    <xs:enumeration
      value="Mms"/>
    <xs:enumeration
      value="Msn"/>
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **PhoneNumberKeyType** simple type:

Value	Meaning
AssistantPhone	Identifies the telephone number as the assistant's telephone number.
BusinessFax	Identifies the telephone number as a business fax number.
BusinessPhone	Identifies the telephone number as a business telephone number.
BusinessPhone2	Identifies the telephone number as a second business telephone number.
Callback	Identifies the telephone number as a callback number.
CarPhone	Identifies the telephone number as a car telephone number.
CompanyMainPhone	Identifies the telephone number as the company's main telephone number.
HomeFax	Identifies the telephone number as a home fax number.

Value	Meaning
HomePhone	Identifies the telephone number as a home telephone number.
HomePhone2	Identifies the telephone number as a second home telephone number.
Isdn	Identifies the telephone number as an Integrated Services Digital Network (ISDN) line.
MobilePhone	Identifies the telephone number as a mobile phone number.
OtherFax	Identifies the telephone number as another fax number.
OtherTelephone	Identifies the telephone number as another telephone number.
Pager	Identifies the telephone number as a pager.
PrimaryPhone	Identifies the telephone number as the primary telephone number.
RadioPhone	Identifies the telephone number as a radio telephone.
Telex	Identifies the telephone number as a telex telephone number.
TtyTddPhone	Identifies the telephone number as a teletype/telecommunication device for the deaf (TTY/TDD) telephone number.
BusinessMobile	Identifies the telephone number as a business mobile phone number.<56>
IPPhone	Voice over IP phone number.<57>
Mms	Not used. <58>
Msn	Not used. <59>

3.1.4.1.2.7   t:PhysicalAddressIndexType Simple Type

The **PhysicalAddressIndexType** simple type identifies the display types for physical addresses.

```
<xs:simpleType name="PhysicalAddressIndexType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="Business"
     />
    <xs:enumeration
      value="Home"
     />
    <xs:enumeration
      value="Other"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **PhysicalAddressIndexType** simple type:

Value	Meaning
None	Indicates that no type is specified for the address.
Business	Displays the address as a business address.
Home	Displays the address as a home address.
Other	Displays the address as an address of type other.

3.1.4.1.2.8   t:PhysicalAddressKeyType Simple Type

The **PhysicalAddressKeyType** simple type identifies the types of physical addresses for a contact (2).

```
<xs:simpleType name="PhysicalAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Business"
     />
    <xs:enumeration
      value="Home"
     />
    <xs:enumeration
      value="Other"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **PhysicalAddressKeyType** simple type:

Value	Meaning
Business	Identifies the address as a business address.
Home	Identifies the address as a home address.
Other	Identifies the address as an address of type other.

3.1.4.2 DeleteItem

This protocol uses the **DeleteItem** operation specified in [MS-OXWSCORE] section 3.1.4.3 to delete contact (2) item elements.

```
<wsdl:operation name="DeleteItem">
  <wsdl:input message="tns:DeleteItemSoapIn" />
  <wsdl:output message="tns:DeleteItemSoapOut" />
</wsdl:operation>
```

Request

Message format	Description
tns:DeleteItemSoapIn ([MS-OXWSCORE] section 3.1.4.3.1.1)	Specifies the SOAP message that defines the contact (2) item to delete. The **DeleteItem** ([MS-OXWSCORE] section 3.1.4.3.2.1) child element that specifies the XML request MUST contain one or more **t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25) elements. All other elements MUST be empty.

Response

Message format	Description
tns:DeleteItemSoapOut ([MS-OXWSCORE] section 3.1.4.3.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.3 UpdateItem

This protocol uses the **UpdateItem** operation specified in [MS-OXWSCORE] section 3.1.4.9 to update contact (2) item elements.

```
<wsdl:operation name="UpdateItem">
  <wsdl:input message="tns:UpdateItemSoapIn" />
  <wsdl:output message="tns:UpdateItemSoapOut" />
</wsdl:operation>
```

Request

Message format	Description
tns:UpdateItemSoapIn ([MS-OXWSCORE] section 3.1.4.9.1.1)	Specifies the SOAP message that defines the contact (2) item to update. The **Items** child element of the **UpdateItem** ([MS-OXWSCORE] section 3.1.4.9.2.1) child element that specifies the XML request MUST contain one or more **t:ContactItemType** elements (section 2.2.4.3). All other elements MUST be empty.

Response

Message format	Description
tns:UpdateItemSoapOut ([MS-OXWSCORE] section 3.1.4.9.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.4 MoveItem

This protocol uses the **MoveItem** operation specified in [MS-OXWSCORE] section 3.1.4.7 to move contact (2) item elements.

```
<wsdl:operation name="MoveItem">
  <wsdl:input message="tns:MoveItemSoapIn" />
  <wsdl:output message="tns:MoveItemSoapOut" />
</wsdl:operation>
```

Request

Message format	Description
tns:MoveItemSoapIn ([MS-OXWSCORE] section 3.1.4.7.1.1)	Specifies the SOAP message that defines the contact (2) item to move. The **Items** child element of the **MoveItem** child element ([MS-OXWSCORE] section 3.1.4.7.2.1) that specifies the XML request MUST contain the following elements: **t:TargetFolderIdType** ([MS-OXWSFOLD] section 2.2.4.16), and **t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25). All other elements MUST be empty.

Response

Message format	Description
tns:MoveItemSoapOut ([MS-OXWSCORE] section 3.1.4.7.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.5 CopyItem

This protocol uses the **CopyItem** operation specified in [MS-OXWSCORE] section 3.1.4.1 to copy contact (2) item elements.

```
<wsdl:operation name="CopyItem">
  <wsdl:input message="tns:CopyItemSoapIn" />
  <wsdl:output message="tns:CopyItemSoapOut" />
</wsdl:operation>
```

Request

Message format	Description
tns:CopyItemSoapIn ([MS-OXWSCORE] section 3.1.4.1.1.1)	Specifies the SOAP message that defines the contact (2) item to copy. The **Items** child element of the **CopyItem** ([MS-OXWSCORE] section 3.1.4.1.2.1) child element that specifies the XML request MUST contain the following elements: **t:TargetFolderIdType** ([MS-OXWSFOLD] section 2.2.4.16), and **t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25). All other elements MUST be empty.

Response

Message format	Description
tns:CopyItemSoapOut ([MS-OXWSCORE] section 3.1.4.1.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.6 CreateItem

This protocol uses the **CreateItem** operation specified in [MS-OXWSCORE] section 3.1.4.2 to create contact (2) item elements.

```
<wsdl:operation name="CreateItem">
    <wsdl:input message="tns:CreateItemSoapIn" />
    <wsdl:output message="tns:CreateItemSoapOut" />
```

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

Request

Message format	Description
tns:CreateItemSoapIn ([MS-OXWSCORE] section 3.1.4.2.1.1)	Specifies the SOAP message that defines the contact (2) item to create. The **Items** child element of the **CreateItem** ([MS-OXWSCORE] section 3.1.4.2.2.1) child element that specifies the XML request MUST contain one or more **t:ContactItemType** elements (section 2.2.4.3). All other elements MUST be empty. The contact (2) item MUST be created in a Contacts folder, or **ErrorCannotCreateContactInNonContactFolder** ([MS-OXWSCDATA] section 2.2.5.24) will be returned.

Response

Message format	Description
tns:CreateItemSoapOut ([MS-OXWSCORE] section 3.1.4.2.1.2)	Specifies the SOAP message returned by the server in response.

3.1.4.7 GetUserPhoto

The **GetUserPhoto WSDL operation** retrieves the profile image for a mailbox.

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

```
<wsdl:operation name="GetUserPhoto">
      <wsdl:input message="tns:GetUserPhotoSoapIn"/>
      <wsdl:output message="tns:GetUserPhotoSoapOut"/>
    </wsdl:operation>
```

The following is the WSDL binding specification of the **GetUserPhoto** WSDL operation.

```
<wsdl:operation name="GetUserPhoto">
      <soap:operation soapAction=
    "http://schemas.microsoft.com/exchange/services/2006/messages/GetUserPhoto"/>
      <wsdl:input>
        <soap:body parts="request" use="literal"/>
        <soap:header message="tns:GetUserPhotoSoapIn"
            part="RequestVersion" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="GetUserPhotoResult" use="literal"/>
        <soap:header message="tns:GetUserPhotoSoapOut"
            part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
```

The protocol client sends a **GetUserPhotoSoapIn** request WSDL message and the protocol server responds with a **GetUserPhotoSoapOut** response WSDL message

3.1.4.7.1 Messages

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

Message	Description
GetUserPhotoSoapIn	Specifies a request to retrieve a photo.
GetUserPhotoSoapOut	Specifies the response to the **GetUserPhotoSoapIn** request WSDL message.

3.1.4.7.1.1   GetUserPhotoSoapIn

The **GetUserPhotoSoapIn** WSDL message specifies a request to retrieve a photo.

The following is the **GetUserPhotoSoapIn** WSDL message specification.

```
<wsdl:message name="GetUserPhotoSoapIn">
    <wsdl:part name="request" element="tns:GetUserPhoto"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>  </wsdl:message>
```

The **GetUserPhotoSoapIn** WSDL message is the input message for the **SOAP action** http://schemas.microsoft.com/exchange/services/2006/messages/GetUserPhoto.

The parts of the **GetUserPhotoSoapIn** WSDL message are described in the following table.

Part name	Element/type	Description
request	**tns:GetUserPhoto** (section 3.1.4.7.2.1)	Specifies the **SOAP body** of the request to retrieve a photo.
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a **SOAP header** that identifies the schema version for the **GetUserPhoto** WSDL operation request.

3.1.4.7.1.2   GetUserPhotoSoapOut

The **GetUserPhotoSoapOut** WSDL message specifies the response to a **GetUserPhotoSoapIn** request WSDL message.

The following is the **GetUserPhotoSoapOut** WSDL message specification.

```
<wsdl:message name="GetUserPhotoSoapOut">
  <wsdl:part name="GetUserPhotoResult" element="tns:GetUserPhotoResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **GetUserPhotoSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/GetUserPhoto.

The parts of the **GetUserPhotoSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
GetUserPhotoResult	**tns:GetUserPhotoResponse** (section	Specifies the SOAP body of the response to

Part name	Element/type	Description
	3.1.4.7.2.2)	a **GetUserPhoto** WSDL operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

A successful **GetUserPhoto** WSDL operation request returns a **GetUserPhotoResponse** element with the **ResponseClass** attribute set to "Success". The **ResponseCode** element of the **GetUserPhotoResponse** element is set to "No Error".

If the **GetUserPhoto** WSDL operation is not successful, it returns a **GetUserPhotoResponse** element with the **ResponseClass** attribute set to "Error". The **ResponseCode** element of the **GetUserPhotoResponse** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.7.2 Elements

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

Element	Description
GetUserPhoto	Specifies the input data for the **GetUserPhoto** WSDL operation.
GetUserPhotoResponse	Specifies the result data for the **GetUserPhoto** WSDL operation.

3.1.4.7.2.1   GetUserPhoto

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

The following is the **GetUserPhoto** element specification.

```
<xs:element name="GetUserPhoto" type="m:GetUserPhotoType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.7.2.2   GetUserPhotoResponse

The **GetUserPhotoResponse** element specifies the result data for the **GetUserPhoto** WSDL operation.

The following is the **GetUserPhotoResponse** element specification.

```
<xs:element name="GetUserPhotoResponse"
        type="m:GetUserPhotoResponseMessageType"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
```

3.1.4.7.3 Complex Types

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

Complex type	Description
GetUserPhotoType	Specifies a request to retrieve a user photo..

Complex type	Description
GetUserPhotoResponseMessageType	Specifies the response message for the **GetUserPhoto** WSDL operation.
GetUserPhotoResponseType	Specifies the response for the **GetUserPhoto** WSDL operation. This complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

3.1.4.7.3.1 GetUserPhotoType

The **GetUserPhotoType** complex type specifies a request to retrieve a user photo. This type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

The following is the **GetUserPhotoType** complex type specification.

```
<xs:complexType name="GetUserPhotoType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="Email" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SizeRequested" type="t:UserPhotoSizeType"
                minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists the child elements of the **GetUserPhotoType** complex type.

Element name	Type	Description
Email	**xs:string** ([XMLSCHEMA2])	Specifies an **email address**.
SizeRequested	**t:UserPhotoSizeType** (section 3.1.4.7.4.1)	Specifies the requested size of the photo.

If the request specifies a size that is not available, the operation returns the largest available photo. If no image is stored on the server, the operation returns the thumbnail image stored in the directory service. The thumbnail image is not necessarily square, even if the size code specifies a square image.

3.1.4.7.3.2 GetUserPhotoResponseMessageType

The **GetUserPhotoResponseMessageType** complex type specifies the response message status for the **GetUserPhoto** request. This type extends the **ResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.65.

The following is the **GetUserPhotoResponseMessageType** complex type specification.

```
<xs:complexType name="GetUserPhotoResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="HasChanged" type="xs:boolean"

                minOccurs="1" maxOccurs="1"/>
```

```
            <xs:element name="PictureData" type="xs:base64Binary"
                   minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

The following table lists the child elements of the **GetUserPhotoResponseMessageType** complex type.

Element name	Type	Description
HasChanged	**xs:boolean** ([XMLSCHEMA2])	Specifies whether the photo has changed.
PictureData	**xs:base64Binary** ([XMLSCHEMA2])	Specifies the binary data for the picture.

3.1.4.7.3.3   GetUserPhotoResponseType

The **GetUserPhotoResponseType** complex type specifies the response for the **GetUserPhoto** WSDL operation. This type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.18.

The following is the **GetUserPhotoResponseType** complex type specification.

```
  <xs:complexType name="GetUserPhotoResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
```

3.1.4.7.4 Simple Types

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

Simple type	Description
UserPhotoSizeType	Specifies the size of the image.

3.1.4.7.4.1   UserPhotoSizeType

Namespace: http://schemas.microsoft.com/exchange/services/2006/types

The **UserPhotoSizeType** simple type specifies the size of the image.

The following is the **UserPhotoSizeType** simple type specification.

```
  <xs:simpleType name="UserPhotoSizeType">
     <xs:restriction base="xs:string">
       <xs:enumeration value="HR48x48" />
       <xs:enumeration value="HR64x64" />
```

```
            <xs:enumeration value="HR96x96" />
            <xs:enumeration value="HR120x120" />
            <xs:enumeration value="HR240x240" />
            <xs:enumeration value="HR360x360" />
            <xs:enumeration value="HR432x432" />
            <xs:enumeration value="HR504x504" />
            <xs:enumeration value="HR648x648" />
        </xs:restriction>
    </xs:simpleType>
    The following table specifies the allowable values for the UserPhotoSizeType simple type.
```

Value	Meaning
HR48x48	The image is 48 pixels high and 48 pixels wide.
HR64x64	The image is 64 pixels high and 64 pixels wide.
HR96x96	The image is 96 pixels high and 96 pixels wide.
HR120x120	The image is 120 pixels high and 120 pixels wide.
HR240x240	The image is 240 pixels high and 240 pixels wide.
HR360x360	The image is 360 pixels high and 360 pixels wide.
HR432x432	The image is 432 pixels high and 432 pixels wide.
HR504x504	The image is 504 pixels high and 504 pixels wide.
HR648x648	The image is 648 pixels high and 648 pixels wide.

3.1.4.7.5 Attributes

None.

3.1.4.7.6 Groups

None.

3.1.4.7.7 Attribute Groups

None.

3.1.4.8 SetUserPhoto

The **SetUserPhoto** WSDL operation adds a photo to a user's account.<60>

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

```
    <wsdl:operation name="SetUserPhoto">
       <wsdl:input message="tns:SetUserPhotoSoapIn"/>
       <wsdl:output message="tns:SetUserPhotoSoapOut"/>
    </wsdl:operation>
```

The following is the WSDL binding specification of the **SetUserPhoto** WSDL operation.

```
      <wsdl:operation name="SetUserPhoto">
       <soap:operation
  soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/SetUserPhoto"/>
        <wsdl:input>
          <soap:header message="tns:SetUserPhotoSoapIn" part="RequestVersion" use="literal"/>
          <soap:body parts="request" use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body parts="SetUserPhotoResult" use="literal"/>
          <soap:header message="tns:SetUserPhotoSoapOut" part="ServerVersion" use="literal"/>
        </wsdl:output>
      </wsdl:operation>
```

The protocol client sends a **SetUserPhotoSoapIn** request WSDL message and the protocol server responds with a **SetUserPhotoSoapOut** response WSDL message

3.1.4.8.1 Messages

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

Message	Description
SetUserPhotoSoapIn	Specifies a request to set a photo in a mailbox.
SetUserPhotoSoapOut	Specifies the response to the **SetUserPhotoSoapIn** request WSDL message.

3.1.4.8.1.1   SetUserPhotoSoapIn

The **SetUserPhotoSoapIn** WSDL message specifies a request to set a photo in a mailbox.

The following is the **SetUserPhotoSoapIn** WSDL message specification.

```
      <wsdl:message name="SetUserPhotoSoapIn">
        <wsdl:part name="request" element="tns:SetUserPhoto"/>
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
      </wsdl:message>
```

The **SetUserPhotoSoapIn** WSDL message is the input message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/SetUserPhoto.

The parts of the **SetUserPhotoSoapIn** WSDL message are described in the following table.

Part name	Element/type	Description
request	**tns:SetUserPhoto** (section 3.1.4.8.2.1)	Specifies the SOAP body of the request to set a photo in a mailbox.
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **SetUserPhoto** WSDL operation request.

3.1.4.8.1.2   SetUserPhotoSoapOut

The **SetUserPhotoSoapOut** WSDL message specifies the response to the **SetUserPhotoSoapIn** request WSDL message.

The following is the **SetUserPhotoSoapOut** WSDL message specification.

```
<wsdl:message name="SetUserPhotoSoapOut">
    <wsdl:part name="SetUserPhotoResult" element="tns:SetUserPhotoResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
```

The **SetUserPhotoSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/SetUserPhoto.

The parts of the **SetUserPhotoSoapOut** WSDL message are described in the following table.

Part name	Element/type	Description
SetUserPhotoResult	**tns:SetUserPhotoResponse** (section 3.1.4.8.2.2)	Specifies the SOAP body of the response to a **SetUserPhoto** WSDL operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.8.2 Elements

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

Element	Description
SetUserPhoto	Specifies the input data for the **SetUserPhoto** WSDL operation.
SetUserPhotoResponse	Specifies the result data for the **SetUserPhoto** WSDL operation.

3.1.4.8.2.1   SetUserPhoto

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

The following is the **SetUserPhoto** element specification.

```
<xs:element name="SetUserPhoto" type="m:SetUserPhotoType"/>
```

3.1.4.8.2.2   SetUserPhotoResponse

The **SetUserPhotoResponse** element specifies the result data for the **SetUserPhoto** WSDL operation.

The following is the **SetUserPhotoResponse** element specification.

```
<xs:element name="SetUserPhotoResponse" type="m:SetUserPhotoResponseMessageType"/>
```

3.1.4.8.3 Complex Types

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

Complex type	Description
SetUserPhotoType	Specifies a request to add a photo to a user account.
SetUserPhotoResponseMessageType	Specifies the response message for the **SetUserPhoto** WSDL operation. This complex type extends the **BaseResponseMessageType** complex type, as specified by [MS-OXWSCDATA] section 2.2.4.18.

3.1.4.8.3.1　SetUserPhotoType

The **SetUserPhotoType** complex type specifies a request to add a photo to a user account. This type extends the **BaseRequestType**, as specified by [MS-OXWSCDATA] section 2.2.4.17.

The following is the **SetUserPhotoType** complex type specification.

```
<xs:complexType name="SetUserPhotoType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="Email" type="t:NonEmptyStringType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="Content" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Email: An element of type **NonEmptyStringType**, as defined in [MS-OXWSCDATA] section 2.2.5.20, that specifies an email address.

Content: An element of base64 **string** type, as defined in [XMLSCHEMA2], that specifies the photo content.

The following table lists the child elements of the **SetUserPhotoType** complex type.

Element name	Type	Description
Email	**t:NonEmptyStringType** ([MS-OXWSCDATA] section 2.2.5.20)	Specifies an email address.
Content	**xs:string** ([XMLSCHEMA2])	Specifies the photo content.

3.1.4.8.3.2　SetUserPhotoResponseMessageType

The **SetUserPhotoResponseMessageType** complex type specifies the response message for the **SetUserPhoto** WSDL operation. This type extends the **BaseResponseMessageType** complex type, as specified by [MS-OXWSCDATA] section 2.2.4.18.

The following is the **SetUserPhotoResponseMessageType** complex type specification.

```
<xs:complexType name="SetUserPhotoResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
< /xs:complexType>
```

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.

3.1.5   Timer Events

None.

3.1.6   Other Local Events

None.

4 Protocol Examples

4.1 Get DateTimeCreated Property

The following is an example of a request to get the **DateTimeCreated** property of an item:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010 SP1"/>
    <t:DateTimePrecision>Milliseconds</t:DateTimePrecision>
  </soap:Header>
  <soap:Body>
    <GetItem xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <ItemShape>
          <t:BaseShape>IdOnly</t:BaseShape>
          <t:AdditionalProperties>
            <t:FieldURI FieldURI="item:DateTimeCreated"/>
          </t:AdditionalProperties>
      </ItemShape>
      <ItemIds>
<!-- The Id of the item being retrieved -->
          <t:ItemId
Id="AQMkADc4ZTU0Y2E4LWE5MjctNDNjOS1hM2QyLTEzMWE4NzhhZjRlOQBGAAAD+TBZmlf7F0ydQQFsX2N8VgcA1ouDJ
nFVkk2sDp7jPV6sQQAAAgEPAAAA1ouDJnFVkk2sDp7jPV6sQQAAAgk1AAAA"/>
      </ItemIds>
    </GetItem>
  </soap:Body>
</soap:Envelope>
```

4.2 Get a User Photo

The following example of the **GetUserPhoto** operation, as described in section , shows how the client retrieves a photo. This example requests a photo 96 pixels high and 96 pixels wide associated with the email address "user1@contoso.com".

```
 <?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
              xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2013"/>
  </soap:Header>
  <soap:Body xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <GetUserPhoto>
        <Email>user1@contoso.com</Email>
        <SizeRequested>HR96x96</SizeRequested>
      </GetUserPhoto>
  </soap:Body>
</soap:Envelope>
```

The server sends the following successful response to the client. The value of the **PictureData** element that contains the returned binary information has been truncated for readability.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="15"
            MinorVersion="0"
            MajorBuildNumber="545"
```

```
                MinorBuildNumber="5"
                Version="Exchange2013"
                xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
                xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    </s:Header>
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetUserPhotoResponse ResponseClass="Success"
            xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
        <ResponseCode>NoError</ResponseCode>
        <HasChanged>true</HasChanged>
        <PictureData>/9jDBkSEw8UHRofHh0aHBwgJC4</PictureData>
      </GetUserPhotoResponse>
    </s:Body>
</s:Envelope>
```

5   Security

5.1   Security Considerations for Implementers

None.

5.2   Index of Security Parameters

None.

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required to implement the functionality that is specified in this document.

File name	Description	Section
MS-OXWSCONT.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSCONT-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7
MS-OXWSCORE-messages.xsd	Contains XML schema message definitions that are referred to by this protocol.	[MS-OXWSCORE] section 7.1

These files have to be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSCONT-types.xsd schema have to be placed in the common folder with these files.

This section contains the content of the MS-OXWSCONT.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
    <wsdl:types>
        <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
            <xs:include schemaLocation="MS-OXWSCORE-messages.xsd"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:portType name="ExchangeServicePortType">
        <wsdl:operation name="GetItem">
            <wsdl:input message="tns:GetItemSoapIn"/>
            <wsdl:output message="tns:GetItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="CreateItem">
            <wsdl:input message="tns:CreateItemSoapIn"/>
            <wsdl:output message="tns:CreateItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="DeleteItem">
            <wsdl:input message="tns:DeleteItemSoapIn"/>
            <wsdl:output message="tns:DeleteItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="UpdateItem">
            <wsdl:input message="tns:UpdateItemSoapIn"/>
            <wsdl:output message="tns:UpdateItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="MoveItem">
            <wsdl:input message="tns:MoveItemSoapIn"/>
            <wsdl:output message="tns:MoveItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="CopyItem">
            <wsdl:input message="tns:CopyItemSoapIn"/>
            <wsdl:output message="tns:CopyItemSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="GetUserPhoto">
            <wsdl:input message="tns:GetUserPhotoSoapIn"/>
            <wsdl:output message="tns:GetUserPhotoSoapOut"/>
        </wsdl:operation>
        <wsdl:operation name="SetUserPhoto">
```

```
                <wsdl:input message="tns:SetUserPhotoSoapIn"/>
                <wsdl:output message="tns:SetUserPhotoSoapOut"/>
            </wsdl:operation>
        </wsdl:portType>
    <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
            <wsdl:documentation>
                <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0"
xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/"/>
            </wsdl:documentation>
            <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
            <wsdl:operation name="GetItem">
                <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetItem"/>
                <wsdl:input>
                    <soap:header message="tns:GetItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:GetItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:GetItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:header message="tns:GetItemSoapIn" part="TimeZoneContext"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
                </wsdl:input>
                <wsdl:output>
                    <soap:body parts="GetItemResult" use="literal"/>
                    <soap:header message="tns:GetItemSoapOut" part="ServerVersion"
use="literal"/>
                </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="CreateItem">
                <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateItem"/>
                <wsdl:input>
                    <soap:header message="tns:CreateItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:CreateItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:CreateItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:header message="tns:CreateItemSoapIn" part="TimeZoneContext"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
                </wsdl:input>
                <wsdl:output>
                    <soap:body parts="CreateItemResult" use="literal"/>
                    <soap:header message="tns:CreateItemSoapOut" part="ServerVersion"
use="literal"/>
                </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="DeleteItem">
                <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteItem"/>
                <wsdl:input>
                    <soap:header message="tns:DeleteItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:DeleteItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:DeleteItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
                </wsdl:input>
                <wsdl:output>
                    <soap:body parts="DeleteItemResult" use="literal"/>
                    <soap:header message="tns:DeleteItemSoapOut" part="ServerVersion"
use="literal"/>
                </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="UpdateItem">
```

```
                    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateItem"/>
                        <wsdl:input>
                            <soap:header message="tns:UpdateItemSoapIn" part="Impersonation"
use="literal"/>
                            <soap:header message="tns:UpdateItemSoapIn" part="MailboxCulture"
use="literal"/>
                            <soap:header message="tns:UpdateItemSoapIn" part="RequestVersion"
use="literal"/>
                            <soap:header message="tns:UpdateItemSoapIn" part="TimeZoneContext"
use="literal"/>
                            <soap:body parts="request" use="literal"/>
                        </wsdl:input>
                        <wsdl:output>
                            <soap:body parts="UpdateItemResult" use="literal"/>
                            <soap:header message="tns:UpdateItemSoapOut" part="ServerVersion"
use="literal"/>
                        </wsdl:output>
                    </wsdl:operation>
                    <wsdl:operation name="MoveItem">
                        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/MoveItem"/>
                        <wsdl:input>
                            <soap:header message="tns:MoveItemSoapIn" part="Impersonation"
use="literal"/>
                            <soap:header message="tns:MoveItemSoapIn" part="MailboxCulture"
use="literal"/>
                            <soap:header message="tns:MoveItemSoapIn" part="RequestVersion"
use="literal"/>
                            <soap:body parts="request" use="literal"/>
                        </wsdl:input>
                        <wsdl:output>
                            <soap:body parts="MoveItemResult" use="literal"/>
                            <soap:header message="tns:MoveItemSoapOut" part="ServerVersion"
use="literal"/>
                        </wsdl:output>
                    </wsdl:operation>
                    <wsdl:operation name="CopyItem">
                        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CopyItem"/>
                        <wsdl:input>
                            <soap:header message="tns:CopyItemSoapIn" part="Impersonation"
use="literal"/>
                            <soap:header message="tns:CopyItemSoapIn" part="MailboxCulture"
use="literal"/>
                            <soap:header message="tns:CopyItemSoapIn" part="RequestVersion"
use="literal"/>
                            <soap:body parts="request" use="literal"/>
                        </wsdl:input>
                        <wsdl:output>
                            <soap:body parts="CopyItemResult" use="literal"/>
                            <soap:header message="tns:CopyItemSoapOut" part="ServerVersion"
use="literal"/>
                        </wsdl:output>
                    </wsdl:operation>
                    <wsdl:operation name="GetUserPhoto">
                     <soap:operation soapAction=
                    "http://schemas.microsoft.com/exchange/services/2006/messages/GetUserPhoto"/>
                      <wsdl:input>
                        <soap:body parts="request" use="literal"/>
                        <soap:header message="tns:GetUserPhotoSoapIn"
                            part="RequestVersion" use="literal"/>
                      </wsdl:input>
                      <wsdl:output>
                        <soap:body parts="GetUserPhotoResult" use="literal"/>
                        <soap:header message="tns:GetUserPhotoSoapOut"
                            part="ServerVersion" use="literal"/>
                      </wsdl:output>
                    </wsdl:operation>
```

```
            <wsdl:operation name="SetUserPhoto">
              <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/SetUserPhoto"/>
              <wsdl:input>
                <soap:header message="tns:SetUserPhotoSoapIn" part="RequestVersion"
use="literal"/>
                <soap:body parts="request" use="literal"/>
              </wsdl:input>
              <wsdl:output>
                <soap:body parts="SetUserPhotoResult" use="literal"/>
                <soap:header message="tns:SetUserPhotoSoapOut" part="ServerVersion"
use="literal"/>
              </wsdl:output>
            </wsdl:operation>
        </wsdl:binding>
        <wsdl:message name="GetItemSoapIn">
            <wsdl:part name="request" element="tns:GetItem"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
            <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
        </wsdl:message>
        <wsdl:message name="GetItemSoapOut">
            <wsdl:part name="GetItemResult" element="tns:GetItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="CreateItemSoapIn">
            <wsdl:part name="request" element="tns:CreateItem"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
            <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
        </wsdl:message>
        <wsdl:message name="CreateItemSoapOut">
            <wsdl:part name="CreateItemResult" element="tns:CreateItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="DeleteItemSoapIn">
            <wsdl:part name="request" element="tns:DeleteItem"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        </wsdl:message>
        <wsdl:message name="DeleteItemSoapOut">
            <wsdl:part name="DeleteItemResult" element="tns:DeleteItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="UpdateItemSoapIn">
            <wsdl:part name="request" element="tns:UpdateItem"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
            <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
        </wsdl:message>
        <wsdl:message name="UpdateItemSoapOut">
            <wsdl:part name="UpdateItemResult" element="tns:UpdateItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="MoveItemSoapIn">
            <wsdl:part name="request" element="tns:MoveItem"/>
            <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
            <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
            <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        </wsdl:message>
        <wsdl:message name="MoveItemSoapOut">
            <wsdl:part name="MoveItemResult" element="tns:MoveItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="CopyItemSoapIn">
```

```
                <wsdl:part name="request" element="tns:CopyItem"/>
                <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
                <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
                <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        </wsdl:message>
        <wsdl:message name="CopyItemSoapOut">
                <wsdl:part name="CopyItemResult" element="tns:CopyItemResponse"/>
                <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="GetUserPhotoSoapIn">
                <wsdl:part name="request" element="tns:GetUserPhoto"/>
                <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        </wsdl:message>
        <wsdl:message name="GetUserPhotoSoapOut">
                <wsdl:part name="GetUserPhotoResult" element="tns:GetUserPhotoResponse"/>
                <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
        <wsdl:message name="SetUserPhotoSoapIn">
                <wsdl:part name="request" element="tns:SetUserPhoto"/>
                <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
        </wsdl:message>
        <wsdl:message name="SetUserPhotoSoapOut">
                <wsdl:part name="SetUserPhotoResult" element="tns:SetUserPhotoResponse"/>
                <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
        </wsdl:message>
    </wsdl:definitions>
```

7 Appendix B: Full XML Schema

For ease of implementation, the following is the full XML schema for this protocol.

This section contains the contents of the MS-OXWSCONT-types.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSCONT-types.xsd includes the file shown in the following table. To operate correctly, this file has to be present in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:complexType name="CompleteNameType">
    <xs:sequence>
      <xs:element name="Title" type="xs:string" minOccurs="0"/>
      <xs:element name="FirstName" type="xs:string" minOccurs="0"/>
      <xs:element name="MiddleName" type="xs:string" minOccurs="0"/>
      <xs:element name="LastName" type="xs:string" minOccurs="0"/>
      <xs:element name="Suffix" type="xs:string" minOccurs="0"/>
      <xs:element name="Initials" type="xs:string" minOccurs="0"/>
      <xs:element name="FullName" type="xs:string" minOccurs="0"/>
      <xs:element name="Nickname" type="xs:string" minOccurs="0"/>
      <xs:element name="YomiFirstName" type="xs:string" minOccurs="0"/>
      <xs:element name="YomiLastName" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ContactItemType">
    <xs:complexContent>
      <xs:extension base="t:ItemType">
        <xs:sequence>
          <xs:element name="FileAs" type="xs:string" minOccurs="0"/>
          <xs:element name="FileAsMapping" type="t:FileAsMappingType" minOccurs="0"/>
          <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
          <xs:element name="GivenName" type="xs:string" minOccurs="0"/>
          <xs:element name="Initials" type="xs:string" minOccurs="0"/>
          <xs:element name="MiddleName" type="xs:string" minOccurs="0"/>
          <xs:element name="Nickname" type="xs:string" minOccurs="0"/>
          <xs:element name="CompleteName" type="t:CompleteNameType" minOccurs="0"/>
          <xs:element name="CompanyName" type="xs:string" minOccurs="0"/>
          <xs:element name="EmailAddresses" type="t:EmailAddressDictionaryType"
minOccurs="0"/>
          <xs:element name="AbchEmailAddresses" type="t:AbchEmailAddressDictionaryType"
minOccurs="0"/>
          <xs:element name="PhysicalAddresses" type="t:PhysicalAddressDictionaryType"
minOccurs="0"/>
          <xs:element name="PhoneNumbers" type="t:PhoneNumberDictionaryType" minOccurs="0"/>
          <xs:element name="AssistantName" type="xs:string" minOccurs="0"/>
          <xs:element name="Birthday" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="BusinessHomePage" type="xs:anyURI" minOccurs="0"/>
          <xs:element name="Children" type="t:ArrayOfStringsType" minOccurs="0"/>
          <xs:element name="Companies" type="t:ArrayOfStringsType" minOccurs="0"/>
          <xs:element name="ContactSource" type="t:ContactSourceType" minOccurs="0"/>
          <xs:element name="Department" type="xs:string" minOccurs="0"/>
          <xs:element name="Generation" type="xs:string" minOccurs="0"/>
          <xs:element name="ImAddresses" type="t:ImAddressDictionaryType" minOccurs="0"/>
```

```
                <xs:element name="JobTitle" type="xs:string" minOccurs="0"/>
                <xs:element name="Manager" type="xs:string" minOccurs="0"/>
                <xs:element name="Mileage" type="xs:string" minOccurs="0"/>
                <xs:element name="OfficeLocation" type="xs:string" minOccurs="0"/>
                <xs:element name="PostalAddressIndex" type="t:PhysicalAddressIndexType"
    minOccurs="0"/>
                <xs:element name="Profession" type="xs:string" minOccurs="0"/>
                <xs:element name="SpouseName" type="xs:string" minOccurs="0"/>
                <xs:element name="Surname" type="xs:string" minOccurs="0"/>
                <xs:element name="WeddingAnniversary" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="HasPicture" type="xs:boolean" minOccurs="0"/>
                <xs:element name="PhoneticFullName" type="xs:string" minOccurs="0" />
                <xs:element name="PhoneticFirstName" type="xs:string" minOccurs="0" />
                <xs:element name="PhoneticLastName" type="xs:string" minOccurs="0" />
                <xs:element name="Alias" type="xs:string" minOccurs="0" />
                <xs:element name="Notes" type="xs:string" minOccurs="0" />
                <xs:element name="Photo" type="xs:base64Binary" minOccurs="0" />
                <xs:element name="UserSMIMECertificate" type="t:ArrayOfBinaryType" minOccurs="0" />
                <xs:element name="MSExchangeCertificate" type="t:ArrayOfBinaryType" minOccurs="0"
    />
                <xs:element name="DirectoryId" type="xs:string" minOccurs="0" />
                <xs:element name="ManagerMailbox" type="t:SingleRecipientType" minOccurs="0" />
                <xs:element name="DirectReports" type="t:ArrayOfRecipientsType" minOccurs="0" />
                <xs:element name="AccountName" type="xs:string" minOccurs="0"/>
                <xs:element name="IsAutoUpdateDisabled" type="xs:boolean" minOccurs="0"/>
                <xs:element name="IsMessengerEnabled" type="xs:boolean" minOccurs="0"/>
                <xs:element name="Comment" type="xs:string" minOccurs="0"/>
                <xs:element name="ContactShortId" type="xs:int" minOccurs="0"/>
                <xs:element name="ContactType" type="xs:string" minOccurs="0"/>
                <xs:element name="Gender" type="xs:string" minOccurs="0"/>
                <xs:element name="IsHidden" type="xs:boolean" minOccurs="0"/>
                <xs:element name="ObjectId" type="xs:string" minOccurs="0"/>
                <xs:element name="PassportId" type="xs:long" minOccurs="0"/>
                <xs:element name="PersonId" type="xs:int" minOccurs="0"/>
                <xs:element name="IsPrivate" type="xs:boolean" minOccurs="0"/>
                <xs:element name="SourceId" type="xs:string" minOccurs="0"/>
                <xs:element name="TrustLevel" type="xs:int" minOccurs="0"/>
                <xs:element name="CreatedBy" type="xs:string" minOccurs="0"/>
                <xs:element name="Urls" type="t:ContactUrlDictionaryType" minOccurs="0"/>
                <xs:element name="Cid" type="xs:long" minOccurs="0"/>
                <xs:element name="SkypeAuthCertificate" type="xs:string" minOccurs="0"/>
                <xs:element name="SkypeContext" type="xs:string" minOccurs="0"/>
                <xs:element name="SkypeId" type="xs:string" minOccurs="0"/>
                <xs:element name="SkypeRelationship" type="xs:string" minOccurs="0"/>
                <xs:element name="YomiNickname" type="xs:string" minOccurs="0"/>
                <xs:element name="XboxLiveTag" type="xs:string" minOccurs="0"/>
                <xs:element name="InviteFree" type="xs:boolean" minOccurs="0"/>
                <xs:element name="HidePresenceAndProfile" type="xs:boolean" minOccurs="0"/>
                <xs:element name="IsPendingOutbound" type="xs:boolean" minOccurs="0"/>
                <xs:element name="SupportGroupFeeds" type="xs:boolean" minOccurs="0"/>
                <xs:element name="UserTileHash" type="xs:string" minOccurs="0"/>
                <xs:element name="UnifiedInbox" type="xs:boolean" minOccurs="0"/>
                <xs:element name="Mris" type="t:ArrayOfStringsType" minOccurs="0"/>
                <xs:element name="Wlid" type="xs:string" minOccurs="0"/>
                <xs:element name="AbchContactId" type="t:GuidType" minOccurs="0"/>
                <xs:element name="NotInBirthdayCalendar" type="xs:boolean" minOccurs="0"/>
                <xs:element name="ShellContactType" type="xs:string" minOccurs="0" />
                <xs:element name ="ImMri" type="xs:int" minOccurs="0" />
                <xs:element name="PresenceTrustLevel" type="xs:string" minOccurs="0" />
                <xs:element name="OtherMri" type="xs:string" minOccurs="0" />
                <xs:element name="ProfileLastChanged" type="xs:string" minOccurs="0" />
                <xs:element name="MobileImEnabled" type="xs:boolean" minOccurs="0" />
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="AbchEmailAddressDictionaryEntryType">
        <xs:sequence>
          <xs:element name="Type" type="t:AbchEmailAddressTypeType" minOccurs="1"/>
```

```xml
            <xs:element name="Address" type="xs:string" minOccurs="1"/>
            <xs:element name="IsMessengerEnabled" type="xs:boolean" minOccurs="0"/>
            <xs:element name="Capabilities" type="xs:long" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="AbchEmailAddressDictionaryType">
        <xs:sequence>
            <xs:element name="Email" type="t:AbchEmailAddressDictionaryEntryType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="AbchEmailAddressTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Personal"/>
            <xs:enumeration value="Business"/>
            <xs:enumeration value="Other"/>
            <xs:enumeration value="Passport"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="AbchPersonContactHandle">
        <xs:sequence>
            <xs:element name="SourceId" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="ObjectId" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="AccountName" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="AbchPersonItemType">
        <xs:complexContent>
            <xs:extension base="t:ItemType">
                <xs:sequence>
                    <xs:element name="AntiLinkInfo" type="xs:string" minOccurs="0"/>
                    <xs:element name="PersonId" type="t:GuidType" minOccurs="0"/>
                    <xs:element name="ContactHandles" type="t:ArrayOfAbchPersonContactHandlesType"
minOccurs="0"/>
                    <xs:element name="ContactCategories" type="t:ArrayOfStringsType" minOccurs="0"/>
                    <xs:element name="RelevanceOrder1" type="xs:string" minOccurs="0"/>
                    <xs:element name="RelevanceOrder2" type="xs:string" minOccurs="0"/>
                    <xs:element name="TrustLevel" type="xs:int" minOccurs="0"/>
                    <xs:element name="FavoriteOrder" type="xs:int" minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="ArrayOfAbchPersonContactHandlesType">
        <xs:sequence>
            <xs:element name="ContactHandle" type="t:AbchPersonContactHandle" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ArrayOfBinaryType">
        <xs:sequence>
            <xs:element name="Base64Binary" type="xs:base64Binary" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ContactsFolderType">
        <xs:complexContent>
            <xs:extension base="t:BaseFolderType">
                <xs:sequence>
                    <xs:element name="SharingEffectiveRights" type="t:PermissionReadAccessType"
minOccurs="0"/>
                    <xs:element name="PermissionSet" type="t:PermissionSetType" minOccurs="0"/>
                    <xs:element name="SourceId" type="xs:string" minOccurs="0"/>
                    <xs:element name="AccountName" type="xs:string" minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:simpleType name="ContactSourceType">
```

```
        <xs:restriction base="xs:string">
          <xs:enumeration value="ActiveDirectory"/>
          <xs:enumeration value="Store"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="ContactUrlDictionaryEntryType">
        <xs:sequence>
          <xs:element name="Type" type="t:ContactUrlKeyType" minOccurs="1"/>
          <xs:element name="Name" type="xs:string" minOccurs="0"/>
          <xs:element name="Address" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="ContactUrlDictionaryType">
        <xs:sequence>
          <xs:element name="Url" type="t:ContactUrlDictionaryEntryType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="ContactUrlKeyType">
        <xs:restriction base="xs:string">
          <xs:enumeration value="Personal"/>
          <xs:enumeration value="Business"/>
          <xs:enumeration value="Attachment"/>
          <xs:enumeration value="EbcDisplayDefinition"/>
          <xs:enumeration value="EbcFinalImage"/>
          <xs:enumeration value="EbcLogo"/>
          <xs:enumeration value="Feed"/>
          <xs:enumeration value="Image"/>
          <xs:enumeration value="InternalMarker"/>
          <xs:enumeration value="Other"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="ContactsViewType">
        <xs:complexContent>
          <xs:extension base="t:BasePagingType">
            <xs:attribute name="InitialName" type="xs:string" use="optional"/>
            <xs:attribute name="FinalName" type="xs:string" use="optional"/>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="EmailAddressDictionaryEntryType">
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="Key" type="t:EmailAddressKeyType" use="required"/>
            <xs:attribute name="Name" type="xs:string" use="optional"/>
            <xs:attribute name="RoutingType" type="xs:string" use="optional"/>
            <xs:attribute name="MailboxType" type="t:MailboxTypeType" use="optional"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
      <xs:complexType name="EmailAddressDictionaryType">
        <xs:sequence>
          <xs:element name="Entry" type="t:EmailAddressDictionaryEntryType"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="EmailAddressKeyType">
        <xs:restriction base="xs:string">
          <xs:enumeration value="EmailAddress1"/>
          <xs:enumeration value="EmailAddress2"/>
          <xs:enumeration value="EmailAddress3"/>
        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="FileAsMappingType">
        <xs:restriction base="xs:string">
          <xs:enumeration value="None"/>
          <xs:enumeration value="LastCommaFirst"/>
          <xs:enumeration value="FirstSpaceLast"/>
          <xs:enumeration value="Company"/>
          <xs:enumeration value="LastCommaFirstCompany"/>
```

```
      <xs:enumeration value="CompanyLastFirst"/>
      <xs:enumeration value="LastFirst"/>
      <xs:enumeration value="LastFirstCompany"/>
      <xs:enumeration value="CompanyLastCommaFirst"/>
      <xs:enumeration value="LastFirstSuffix"/>
      <xs:enumeration value="LastSpaceFirstCompany"/>
      <xs:enumeration value="CompanyLastSpaceFirst"/>
      <xs:enumeration value="LastSpaceFirst"/>
      <xs:enumeration value="DisplayName"/>
      <xs:enumeration value="FirstName"/>
      <xs:enumeration value="LastFirstMiddleSuffix"/>
      <xs:enumeration value="LastName"/>
      <xs:enumeration value="Empty"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ImAddressDictionaryEntryType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Key" type="t:ImAddressKeyType" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="ImAddressDictionaryType">
    <xs:sequence>
      <xs:element name="Entry" type="t:ImAddressDictionaryEntryType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ImAddressKeyType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ImAddress1"/>
      <xs:enumeration value="ImAddress2"/>
      <xs:enumeration value="ImAddress3"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="PhoneNumberDictionaryEntryType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Key" type="t:PhoneNumberKeyType" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="PhoneNumberDictionaryType">
    <xs:sequence>
      <xs:element name="Entry" type="t:PhoneNumberDictionaryEntryType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="PhoneNumberKeyType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AssistantPhone"/>
      <xs:enumeration value="BusinessFax"/>
      <xs:enumeration value="BusinessPhone"/>
      <xs:enumeration value="BusinessPhone2"/>
      <xs:enumeration value="Callback"/>
      <xs:enumeration value="CarPhone"/>
      <xs:enumeration value="CompanyMainPhone"/>
      <xs:enumeration value="HomeFax"/>
      <xs:enumeration value="HomePhone"/>
      <xs:enumeration value="HomePhone2"/>
      <xs:enumeration value="Isdn"/>
      <xs:enumeration value="MobilePhone"/>
      <xs:enumeration value="OtherFax"/>
      <xs:enumeration value="OtherTelephone"/>
      <xs:enumeration value="Pager"/>
      <xs:enumeration value="PrimaryPhone"/>
      <xs:enumeration value="RadioPhone"/>
      <xs:enumeration value="Telex"/>
      <xs:enumeration value="TtyTddPhone"/>
      <xs:enumeration value="BusinessMobile"/>
```

```
              <xs:enumeration value="IPPhone"/>
              <xs:enumeration value="Mms"/>
              <xs:enumeration value="Msn"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="PhysicalAddressDictionaryEntryType">
        <xs:sequence>
            <xs:element name="Street" type="xs:string" minOccurs="0"/>
            <xs:element name="City" type="xs:string" minOccurs="0"/>
            <xs:element name="State" type="xs:string" minOccurs="0"/>
            <xs:element name="CountryOrRegion" type="xs:string" minOccurs="0"/>
            <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="Key" type="t:PhysicalAddressKeyType" use="required"/>
    </xs:complexType>
    <xs:complexType name="PhysicalAddressDictionaryType">
        <xs:sequence>
            <xs:element name="Entry" type="t:PhysicalAddressDictionaryEntryType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="PhysicalAddressIndexType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="None"/>
            <xs:enumeration value="Business"/>
            <xs:enumeration value="Home"/>
            <xs:enumeration value="Other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="PhysicalAddressKeyType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Business"/>
            <xs:enumeration value="Home"/>
            <xs:enumeration value="Other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="GetUserPhotoType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:sequence>
                    <xs:element name="Email" type="xs:string" minOccurs="1" maxOccurs="1"/>
                    <xs:element name="SizeRequested" type="t:UserPhotoSizeType"
                            minOccurs="1" maxOccurs="1"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="GetUserPhoto" type="m:GetUserPhotoType"/>
    <xs:complexType name="GetUserPhotoResponseMessageType">
        <xs:complexContent>
            <xs:extension base="m:ResponseMessageType">
                <xs:sequence>
                    <xs:element name="HasChanged" type="xs:boolean"
                            minOccurs="1" maxOccurs="1"/>
                    <xs:element name="PictureData" type="xs:base64Binary"
                            minOccurs="0" maxOccurs="1"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="GetUserPhotoResponseType">
        <xs:complexContent>
            <xs:extension base="m:BaseResponseMessageType"/>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="GetUserPhotoResponse"
            type="m:GetUserPhotoResponseMessageType"/>
    <xs:complexType name="SetUserPhotoType">
        <xs:complexContent>
```

```
            <xs:extension base="m:BaseRequestType">
              <xs:sequence>
                <xs:element name="Email" type="t:NonEmptyStringType" minOccurs="1" maxOccurs="1"/>
                <xs:element name="Content" type="xs:string" minOccurs="1" maxOccurs="1"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
        <xs:element name="SetUserPhoto" type="m:SetUserPhotoType"/>
        <xs:complexType name="SetUserPhotoResponseMessageType">
          <xs:complexContent>
            <xs:extension base="m:BaseResponseMessageType"/>
          </xs:complexContent>
        </xs:complexType>
        <xs:element name="SetUserPhotoResponse" type="m:SetUserPhotoResponseMessageType"/>
        <xs:simpleType name="UserPhotoSizeType">
          <xs:restriction base="xs:string">
            <xs:enumeration value="HR48x48" />
            <xs:enumeration value="HR64x64" />
            <xs:enumeration value="HR96x96" />
            <xs:enumeration value="HR120x120" />
            <xs:enumeration value="HR240x240" />
            <xs:enumeration value="HR360x360" />
            <xs:enumeration value="HR432x432" />
            <xs:enumeration value="HR504x504" />
            <xs:enumeration value="HR648x648" />
          </xs:restriction>
        </xs:simpleType>
      </xs:schema>
```

8   Appendix C: 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 Exchange Server 2007

- Microsoft Exchange Server 2010

- Microsoft Exchange Server 2013

- Microsoft Exchange Server 2016

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 2.2.4.1: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchPersonItemType** complex type.

<2> Section 2.2.4.2: Exchange 2007, Exchange 2010, and Microsoft Exchange Server 2010 Service Pack 1 (SP1) do not include the **ArrayOfBinaryType** complex type. This type was introduced in Microsoft Exchange Server 2010 Service Pack 2 (SP2).

<3> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2010 SP1 do not include the following elements: **PhoneticFullName**, **PhoneticFirstName**, **PhoneticLastName**, **Alias**, **Notes**, **Photo**, **UserSMIMECertificate**, **MSExchangeCertificate**, **DirectoryId**, **ManagerMailbox**, and **DirectReports**. These elements were introduced in Exchange 2010 SP2.

<4> Section 2.2.4.3: Exchange 2007 does not support the **HasPicture** element.

<5> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **AccountName** element.

<6> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **IsAutoUpdateDisabled** element.

<7> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Comment** element.

<8> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **ContactShortId** element.

<9> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **ContactType** element.

<10> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Gender** element.

<11> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **IsHidden** element.

<12> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **ObjectId** element.

<13> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **PassportId** element.

<14> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **PersonId** element.

<15> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **IsPrivate** element.

<16> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SourceId** element.

<17> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **TrustLevel** element.

<18> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **CreatedBy** element.

<19> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Urls** element.

<20> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Cid** element.

<21> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SkypeAuthCertificate** element.

<22> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SkypeContext** element.

<23> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SkypeId** element.

<24> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SkypeRelationship** element.

<25> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **YomiNickname** element.

<26> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **XboxLiveTag** element.

<27> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **InviteFree** element.

<28> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **HidePresenceAndProfile** element.

<29> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **IsPendingOutbound** element.

<30> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **SupportGroupFeeds** element.

<31> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **UserTileHash** element.

<32> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **UnifiedInbox** element.

<33> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Mris** element.

<34> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Wlid** element.

<35> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **AbchContactId** element.

<36> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **NotInBirthdayCalendar** element.

<37> Section 3.1.4.1.1.1: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchEmailAddressDictionaryEntryType** complex type.

<38> Section 3.1.4.1.1.2: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchEmailAddressDictionaryType** complex type.

<39> Section 3.1.4.1.1.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchPersonContactHandle** complex type.

<40> Section 3.1.4.1.1.4: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ArrayOfAbchPersonContactHandlesType** complex type.

<41> Section 3.1.4.1.1.6: Exchange 2007 does not support the **SharingEffectiveRights** element.

<42> Section 3.1.4.1.1.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **SourceId** element.

<43> Section 3.1.4.1.1.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AccountName** element.

<44> Section 3.1.4.1.1.8: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ContactUrlDictionaryEntryType** complex type.

<45> Section 3.1.4.1.1.9: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ContactUrlDictionaryType** complex type

<46> Section 3.1.4.1.1.10: Exchange 2007 does not support the **Name** attribute.

<47> Section 3.1.4.1.1.10: Exchange 2007 does not support the **RoutingType** attribute.

<48> Section 3.1.4.1.1.10: Exchange 2007 does not support the **MailboxType** attribute.

<49> Section 3.1.4.1.2.1: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchEmailAddressTypeType** simple type.

<50> Section 3.1.4.1.2.2: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ContactUrlKeyType** simple type.

<51> Section 3.1.4.1.2.4: Exchange 2007 does not support the **DisplayName** attribute.

<52> Section 3.1.4.1.2.4: Exchange 2007 does not support the **FirstName** attribute.

<53> Section 3.1.4.1.2.4: Exchange 2007 does not support the **LastFirstMiddleSuffix** attribute.

<54> Section 3.1.4.1.2.4: Exchange 2007 does not support the **LastName** attribute.

<55> Section 3.1.4.1.2.4: Exchange 2007 does not support the **Empty** attribute.

<56> Section 3.1.4.1.2.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **BusinessMobile** value.

<57> Section 3.1.4.1.2.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **IPPhone** value.

<58> Section 3.1.4.1.2.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Mms** value.

<59> Section 3.1.4.1.2.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not use the **Msn** value.

<60> Section 3.1.4.8: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **SetUserPhoto** operation.

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

▪ A document revision that incorporates changes to interoperability requirements or functionality.

▪ The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

▪ New content added.

▪ Content updated.

▪ Content removed.

▪ New product behavior note added.

▪ Product behavior note updated.

▪ Product behavior note removed.

▪ New protocol syntax added.

▪ Protocol syntax updated.

▪ Protocol syntax removed.

▪ New content added due to protocol revision.

▪ Content updated due to protocol revision.

▪ Content removed due to protocol revision.

▪ New protocol syntax added due to protocol revision.

▪ Protocol syntax updated due to protocol revision.

▪ Protocol syntax removed due to protocol revision.

▪ Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.1 Namespaces	Added namespace prefix m.	N	Content update.
2.2.4.1 t:AbchPersonItemType Complex Type	Added a new section for this complex type.	Y	New content added.
2.2.4.1 t:AbchPersonItemType Complex Type	Added a new product behavior note for this complex type.	N	New product behavior note added.
2.2.4.3 t:ContactItemType Complex Type	Added descriptions for several child elements for this complex type.	N	Content update.
2.2.4.3 t:ContactItemType Complex Type	Added product behavior notes for several child elements for this complex type.	N	New product behavior note added.
3.1.4.1.1.1 t:AbchEmailAddressDictionaryEntryType Complex Type	Added a product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.1 t:AbchEmailAddressDictionaryEntryType Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.1.2 t:AbchEmailAddressDictionaryType Complex Type	Added a product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.2 t:AbchEmailAddressDictionaryType Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.1.3 t:AbchPersonContactHandle Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.1.3 t:AbchPersonContactHandle Complex Type	Added a product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.4 t:ArrayOfAbchPersonContactHandlesType Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.1.4 t:ArrayOfAbchPersonContactHandlesType Complex Type	Added a new product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.6 t:ContactsFolderType Complex Type	Added details for the SourceId and AccountName child	N	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	elements.		
3.1.4.1.1.6 t:ContactsFolderType Complex Type	Added product behavior notes for the SourceId and AccountName child elements.	N	New product behavior note added.
3.1.4.1.1.8 t:ContactUrlDictionaryEntryType Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.1.8 t:ContactUrlDictionaryEntryType Complex Type	Added a new product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.9 t:ContactUrlDictionaryType Complex Type	Added a new product behavior note for this complex type.	N	New product behavior note added.
3.1.4.1.1.9 t:ContactUrlDictionaryType Complex Type	Added a new section for this complex type.	Y	New content added.
3.1.4.1.2.1 t:AbchEmailAddressTypeType Simple Type	Added a new section for this simple type.	Y	New content added.
3.1.4.1.2.1 t:AbchEmailAddressTypeType Simple Type	Added a new product behavior note for this simple type.	N	New product behavior note added.
3.1.4.1.2.2 t:ContactUrlKeyType Simple Type	Added a new section for this simple type.	Y	New content added.
3.1.4.1.2.2 t:ContactUrlKeyType Simple Type	Added a new product behavior note for this simple type.	N	New product behavior note added.
3.1.4.1.2.6 t:PhoneNumberKeyType Simple Type	Added details for four new values for this simple type: BusinessMobile, IPPhone, Mms, and Msn.	N	New content added.
3.1.4.1.2.6 t:PhoneNumberKeyType Simple Type	Added product behavior notes for four new values for this simple type: BusinessMobile, IPPhone, Mms, and Msn.	N	New product behavior note added.
3.1.4.7 GetUserPhoto	Added new section for this operation and its associated messages, elements, complex types, and simple types.	Y	New content added.
3.1.4.8 SetUserPhoto	Added new section for this operation and its associated messages, elements, complex types, and simple types.	Y	New content added.
3.1.4.8 SetUserPhoto	Added a new product behavior note for this operation.	N	New product behavior note added.
4.1 Get DateTimeCreated Property	Updated example schema.	N	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
4.2 Get a User Photo	Added a new section for this example.	Y	New content added.
7 Appendix B: Full XML Schema	Updated namespace prefixes tns and m.	N	Content update.
8 Appendix C: Product Behavior	Updated the list of applicable products.	Y	Content update.

10 Index