Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/15/2009</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Availability.</td>
</tr>
<tr>
<td>11/4/2009</td>
<td>1.1.0</td>
<td>Minor</td>
<td>Updated the technical content.</td>
</tr>
<tr>
<td>2/10/2010</td>
<td>2.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>5/5/2010</td>
<td>2.0.1</td>
<td>Editorial</td>
<td>Revised and edited the technical content.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>3.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>3.0</td>
<td>None</td>
<td>No changes to the meaning, language, and formatting of the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>3.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>3.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>4.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>4.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>4.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>4.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>5.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>6.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>7.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>7.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>8.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>10.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>11.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>12.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/19/2017</td>
<td>13.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>12/12/2017</td>
<td>13.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>14.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>15.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/24/2019</td>
<td>15.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>12/4/2019</td>
<td>15.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
</tbody>
</table>
# Table of Contents

## 1 Introduction ........................................................................................................... 7
  1.1 Glossary .................................................................................................................. 7
  1.2 References .............................................................................................................. 9
    1.2.1 Normative References ..................................................................................... 9
    1.2.2 Informative References .................................................................................... 9
  1.3 Overview ................................................................................................................. 10
  1.4 Relationship to Other Protocols ............................................................................. 10
  1.5 Prerequisites/Preconditions ................................................................................... 10
  1.6 Applicability Statement ......................................................................................... 10
  1.7 Versioning and Capability Negotiation ................................................................. 10
  1.8 Vendor-Extensible Fields ...................................................................................... 11
  1.9 Standards Assignments ......................................................................................... 11

## 2 Messages .................................................................................................................. 12
  2.1 Transport ................................................................................................................. 12
  2.2 Common Message Syntax ...................................................................................... 12
    2.2.1 Namespaces ..................................................................................................... 12
    2.2.2 Messages ......................................................................................................... 12
    2.2.3 Elements .......................................................................................................... 12
    2.2.4 Complex Types ............................................................................................... 13
      2.2.4.1 $t$:AbchPersonItemType Complex Type................................................. 13
      2.2.4.2 $t$:ArrayOfBinaryType Complex Type............................................... 14
      2.2.4.3 $t$:ContactItemType Complex Type...................................................... 14
    2.2.5 Simple Types .................................................................................................... 23
      2.2.5.1 $t$:ContactSourceType Simple Type..................................................... 23
    2.2.6 Attributes ......................................................................................................... 24
    2.2.7 Groups ............................................................................................................. 24
    2.2.8 Attribute Groups ............................................................................................. 24

## 3 Protocol Details ..................................................................................................... 25
  3.1 ExchangeServicePortType Server Details ............................................................. 25
    3.1.1 Abstract Data Model ....................................................................................... 25
    3.1.2 Timers ............................................................................................................. 25
    3.1.3 Initialization ..................................................................................................... 25
    3.1.4 Message Processing Events and Sequencing Rules ........................................... 25
      3.1.4.1 GetItem ..................................................................................................... 25
        3.1.4.1.1 Complex Types .................................................................................... 26
          3.1.4.1.1.1 $t$:AbchEmailAddressDictionaryEntryType Complex Type........ 26
          3.1.4.1.1.2 $t$:AbchEmailAddressDictionaryType Complex Type............... 27
          3.1.4.1.1.3 $t$:AbchPersonContactHandle Complex Type.......................... 27
          3.1.4.1.1.4 $t$:ArrayOfAbchPersonContactHandlesType Complex Type........ 27
          3.1.4.1.1.5 $t$:CompleteNameType Complex Type..................................... 28
          3.1.4.1.1.6 $t$:ContactsFolderType Complex Type....................................... 29
          3.1.4.1.1.7 $t$:ContactsViewType Complex Type......................................... 30
          3.1.4.1.1.8 $t$:ContactUrlDictionaryEntryType Complex Type.................. 30
          3.1.4.1.1.9 $t$:ContactUrlDictionaryType Complex Type............................ 31
          3.1.4.1.1.10 $t$:EmailAddressDictionaryEntryType Complex Type.............. 31
          3.1.4.1.1.11 $t$:EmailAddressDictionaryType Complex Type...................... 32
          3.1.4.1.1.12 $t$:ImAddressDictionaryEntryType Complex Type.................. 33
          3.1.4.1.1.13 $t$:ImAddressDictionaryType Complex Type............................ 33
          3.1.4.1.1.14 $t$:PhoneNumberDictionaryEntryType Complex Type.............. 33
          3.1.4.1.1.15 $t$:PhoneNumberDictionaryType Complex Type....................... 34
          3.1.4.1.1.16 $t$:PhysicalAddressDictionaryEntryType Complex Type........... 34
          3.1.4.1.1.17 $t$:PhysicalAddressDictionaryType Complex Type.................. 35
        3.1.4.1.2 Simple Types ......................................................................................... 36
1 Introduction

This document specifies the Contacts Web Service protocol.

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

1.1 Glossary

This document uses the following terms:

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, 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.

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

e-mail 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].

instant messaging: A method of real-time communication over the Internet in which a sender types a message to one or more recipients and the recipient immediately receives the message in a pop-up window.

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-OXWSLIST] 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".


1.2.2 Informative References


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.

![Layering Diagram](image)

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.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>soap</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/soap/">http://schemas.xmlsoap.org/wsd1/soap/</a></td>
<td>[SOAP1.1]</td>
</tr>
<tr>
<td>tns</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td></td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>[XMLSCHEMA2]</td>
</tr>
<tr>
<td>wsdI</td>
<td><a href="http://schemas.xmlsoap.org/wsd1/">http://schemas.xmlsoap.org/wsd1/</a></td>
<td>[WSDL]</td>
</tr>
<tr>
<td>t</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td><a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a></td>
<td></td>
</tr>
<tr>
<td>wsi</td>
<td><a href="http://ws-i.org/schemas/conformanceClaim/">http://ws-i.org/schemas/conformanceClaim/</a></td>
<td>[WSIBASIC]</td>
</tr>
</tbody>
</table>

2.2.2 Messages

This specification does not define any common WSDL message definitions.

2.2.3 Elements

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

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

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>t:AbchPersonItemType (section 2.2.4.1)</td>
<td>Specifies a person.</td>
</tr>
<tr>
<td>t:ArrayOfBinaryType (section 2.2.4.2)</td>
<td>Specifies a collection of certificates for a contact (2).</td>
</tr>
<tr>
<td>t:ContactItemType (section 2.2.4.3)</td>
<td>Represents a server contact (2) item.</td>
</tr>
</tbody>
</table>

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.

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

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersonIdGuid</td>
<td>t:GuidType ([MS-OXWSXPROP] section 2.1.7)</td>
<td>A GUID specifies the ID of this person.&lt;2&gt;</td>
</tr>
<tr>
<td>PersonId</td>
<td>xs:int ([XMLSCHEMA2])</td>
<td>The ID of this person.</td>
</tr>
<tr>
<td>FavoriteOrder</td>
<td>xs:int</td>
<td>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.</td>
</tr>
<tr>
<td>TrustLevel</td>
<td>xs:int</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>RelevanceOrder1</td>
<td>xs:string ([XMLSCHEMA2])</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>RelevanceOrder2</td>
<td>xs:string</td>
<td>For internal use only.</td>
</tr>
</tbody>
</table>
### Element | Type | Description
--- | --- | ---
AntiLinkInfo | xs:string | An ID of a set of people who MUST NOT be linked together automatically.
ContactCategories | t:ArrayOfStringsType | The categories of groups that this person belongs to.
ContactHandles | t:ArrayOfAbchPersonContactHandlesType | A list of the handles to contacts represented by this person.
ExchangePersonIdGuid | t:GuidType | A GUID specifies the person ID.<3>

### 2.2.4.2 t:ArrayOfBinaryType Complex Type

The **ArrayOfBinaryType** complex type specifies a collection of certificates for a contact (2).<4>

This type is used by the **UserSMIMECertificate** element and the **MExchangeCertificate** 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>
```

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base64Binary</td>
<td>xs:base64Binary</td>
<td>Specifies a single certificate for a contact (2). The value is encoded with base64 encoding.</td>
</tr>
</tbody>
</table>

### 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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```
<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="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="Photo"
  type="xs:base64Binary"
  minOccurs="0">
</xs:element>

<xs:element name="UserSMIMECertificate"
  type="t:ArrayOfBinaryType"
  minOccurs="0">
</xs:element>

<xs:element name="MSExchangeCertificate"
  type="t:ArrayOfBinaryType"
  minOccurs="0">
</xs:element>

<xs:element name="DirectoryId"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="ManagerMailbox"
  type="t:SingleRecipientType"
  minOccurs="0">
</xs:element>

<xs:element name="DirectReports"
  type="t:ArrayOfRecipientsType"
  minOccurs="0">
</xs:element>

<xs:element name="AccountName"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="IsAutoUpdateDisabled"
  type="xs:boolean"
  minOccurs="0">
</xs:element>

<xs:element name="IsMessengerEnabled"
  type="xs:boolean"
  minOccurs="0">
</xs:element>

<xs:element name="Comment"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="ContactShortId"
  type="xs:int"
  minOccurs="0">
</xs:element>

<xs:element name="ContactType"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="Gender"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="IsHidden"
  type="xs:boolean"
  minOccurs="0">
</xs:element>

<xs:element name="ObjectId"
  type="xs:string"
  minOccurs="0">
</xs:element>

<xs:element name="PassportId"
  type="xs:long"
  minOccurs="0">
</xs:element>

<xs:element name="IsPrivate"
  type="xs:boolean"
<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="AbchEmailAddressDictionaryType"
    minOccurs="0" type="t:AbchEmailAddressDictionaryType"/>
<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"
<xs:sequence>
  <xs:element name="DisplayNamePrefix" type="xs:string" minOccurs="0" />
  <xs:element name="YomiGivenName" type="xs:string" minOccurs="0" />
  <xs:element name="YomiSurname" type="xs:string" minOccurs="0" />
  <xs:element name="PersonalNotes" type="xs:string" minOccurs="0" />
  <xs:element name="PersonId" type="t:ItemIdType" minOccurs="0" />
</xs:sequence>
</xs:extension>
Child Elements <5>

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileAs</td>
<td>xs:string [XMLSCHEMA2]</td>
<td>Represents how a contact (2) is filed in the Contacts folder.</td>
</tr>
<tr>
<td>FileAsMapping</td>
<td>t:FileAsMappingType (section 3.1.4.1.2.4)</td>
<td>Defines how to construct what is displayed for a contact (2).</td>
</tr>
<tr>
<td>DisplayName</td>
<td>xs:string</td>
<td>Contains the display name of a contact (2).</td>
</tr>
<tr>
<td>GivenName</td>
<td>xs:string</td>
<td>Contains the given name for a contact (2).</td>
</tr>
<tr>
<td>Initials</td>
<td>xs:string</td>
<td>Contains the initials for a contact (2).</td>
</tr>
<tr>
<td>MiddleName</td>
<td>xs:string</td>
<td>Represents the middle name of a contact (2).</td>
</tr>
<tr>
<td>Nickname</td>
<td>xs:string</td>
<td>Represents the nickname of a contact (2).</td>
</tr>
<tr>
<td>CompleteName</td>
<td>t:CompleteNameType (section 3.1.4.1.1.5)</td>
<td>Represents the complete name of a contact (2). This property is read-only</td>
</tr>
<tr>
<td>For client.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompanyName</td>
<td>xs:string</td>
<td>Contains the company name that is associated with a contact (2).</td>
</tr>
<tr>
<td>EmailAddresses</td>
<td>t:EmailAddressDictionaryType (section 3.1.4.1.1.11)</td>
<td>Contains email addresses that are associated with a contact (2).</td>
</tr>
<tr>
<td>PhysicalAddresses</td>
<td>t:PhysicalAddressDictionaryType (section 3.1.4.1.1.17)</td>
<td>Represents a collection of physical addresses that are associated with a contact (2).</td>
</tr>
<tr>
<td>PhoneNumbers</td>
<td>t:PhoneNumberDictionaryType (section 3.1.4.1.1.15)</td>
<td>Represents a collection of telephone numbers for a contact (2).</td>
</tr>
<tr>
<td>AssistantName</td>
<td>xs:string</td>
<td>Contains the name of the assistant for the contact (2).</td>
</tr>
<tr>
<td>Birthday</td>
<td>xs:dateTime [XMLSCHEMA2]</td>
<td>Represents the birthday of the contact (2).</td>
</tr>
<tr>
<td>BusinessHomePage</td>
<td>xs:anyURI [XMLSCHEMA2]</td>
<td>Contains the business home page Uniform Resource Identifier (URI) of a contact (2).</td>
</tr>
<tr>
<td>Companies</td>
<td>t:ArrayOfStringsType ([MS-OWWSCDATA] section 2.2.4.13)</td>
<td>Contains the names of companies that are associated with a contact (2).</td>
</tr>
<tr>
<td>ContactSource</td>
<td>t:ContactSourceType (section 2.2.5.1)</td>
<td>Describes whether the contact (2) is located in the server message store or the directory service.</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Department</td>
<td>xs:string</td>
<td>Contains the work department for the contact (2).</td>
</tr>
<tr>
<td>Generation</td>
<td>xs:string</td>
<td>Contains a generational abbreviation that follows the full name of a contact (2).</td>
</tr>
<tr>
<td>ImAddresses</td>
<td>t:ImAddressDictionaryType (section 3.1.4.1.1.13)</td>
<td>Contains instant messaging addresses for a contact (2).</td>
</tr>
<tr>
<td>JobTitle</td>
<td>xs:string</td>
<td>Contains the job title of a contact (2).</td>
</tr>
<tr>
<td>Manager</td>
<td>xs:string</td>
<td>Represents the manager of a contact (2).</td>
</tr>
<tr>
<td>Mileage</td>
<td>xs:string</td>
<td>Represents the mileage for a contact (2).</td>
</tr>
<tr>
<td>OfficeLocation</td>
<td>xs:string</td>
<td>Represents the office location of a contact (2).</td>
</tr>
<tr>
<td>PostalAddressIndex</td>
<td>t:PhysicalAddressIndexType (section 3.1.4.1.2.7)</td>
<td>Represents the index of one of the physical addresses, which is a contact's (2) mailing address.</td>
</tr>
<tr>
<td>Profession</td>
<td>xs:string</td>
<td>Represents the profession of a contact (2).</td>
</tr>
<tr>
<td>SpouseName</td>
<td>xs:string</td>
<td>Represents the name of the spouse/partner of a contact (2).</td>
</tr>
<tr>
<td>Surname</td>
<td>xs:string</td>
<td>Contains the surname of a contact (2).</td>
</tr>
<tr>
<td>WeddingAnniversary</td>
<td>xs:dateTime</td>
<td>Contains the wedding anniversary date of a contact (2).</td>
</tr>
<tr>
<td>HasPicture</td>
<td>xs:boolean [XMLSCHEMA2]</td>
<td>Represents that the contact (2) has a picture. This element is read-only for the client. &lt;6&gt;</td>
</tr>
<tr>
<td>PhoneticFullName</td>
<td>xs:string</td>
<td>Contains the full name of a contact (2), including the first and last name, spelled phonetically.</td>
</tr>
<tr>
<td>PhoneticFirstName</td>
<td>xs:string</td>
<td>Contains the first name of a contact (2), spelled phonetically.</td>
</tr>
<tr>
<td>PhoneticLastName</td>
<td>xs:string</td>
<td>Contains the last name of a contact (2), spelled phonetically.</td>
</tr>
<tr>
<td>Alias</td>
<td>xs:string</td>
<td>Contains the email alias of a contact.</td>
</tr>
<tr>
<td>Notes</td>
<td>xs:string</td>
<td>Contains supplementary contact (2) information.</td>
</tr>
<tr>
<td>Photo</td>
<td>xs:base64Binary [XMLSCHEMA2]</td>
<td>Contains a value that encodes the photo of a contact (2).</td>
</tr>
<tr>
<td>UserSMIMECertificate</td>
<td>t:ArrayOfBinaryType (section 2.2.4.2)</td>
<td>Contains a value that encodes a contacts (2) S/MIME certificate.</td>
</tr>
<tr>
<td>MSExchangeCertificate</td>
<td>t:ArrayOfBinaryType</td>
<td>Contains a value that encodes the server certificate of a contact (2).</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DirectoryId</td>
<td>xs:string</td>
<td>Contains the directory identifier of a contact. (2).</td>
</tr>
<tr>
<td>ManagerMailbox</td>
<td>t:SingleRecipientType ([MS-OXWSCDATA] section 2.2.4.71)</td>
<td>Contains SMTP information that identifies the mailbox of a contact’s (2) manager.</td>
</tr>
<tr>
<td>DirectReports</td>
<td>t:ArrayOfRecipientsType ([MS-OXWSCDATA] section 2.2.4.11)</td>
<td>Contains SMTP information that identifies the mailboxes of a contact’s (2) direct reports.</td>
</tr>
<tr>
<td>AccountName</td>
<td>xs:string</td>
<td>The account name. &lt;7&gt;</td>
</tr>
<tr>
<td>IsAutoUpdateDisabled</td>
<td>xs:boolean</td>
<td>Specifies whether auto-update is disabled for this contact. &lt;8&gt;</td>
</tr>
<tr>
<td>IsMessengerEnabled</td>
<td>xs:boolean</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>Comment</td>
<td>xs:string</td>
<td>A comment. &lt;9&gt;</td>
</tr>
<tr>
<td>ContactShortId</td>
<td>xs:int</td>
<td>For internal use only. &lt;10&gt;</td>
</tr>
<tr>
<td>ContactType</td>
<td>xs:string</td>
<td>The type of the contact. &lt;11&gt;</td>
</tr>
<tr>
<td>Gender</td>
<td>xs:string</td>
<td>The gender of the contact. &lt;12&gt;</td>
</tr>
<tr>
<td>IsHidden</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;13&gt;</td>
</tr>
<tr>
<td>ObjectId</td>
<td>xs:string</td>
<td>The handle of the contact. &lt;14&gt;</td>
</tr>
<tr>
<td>PassportId</td>
<td>xs:long</td>
<td>For internal use only. &lt;15&gt;</td>
</tr>
<tr>
<td>IsPrivate</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;16&gt;</td>
</tr>
<tr>
<td>SourceId</td>
<td>xs:string</td>
<td>Specifies the source of the contact. &lt;17&gt;</td>
</tr>
<tr>
<td>TrustLevel</td>
<td>xs:int</td>
<td>For internal use only. &lt;18&gt;</td>
</tr>
<tr>
<td>CreatedBy</td>
<td>xs:string</td>
<td>For internal use only. &lt;19&gt;</td>
</tr>
<tr>
<td>Uris</td>
<td>t:ContactUrlDictionaryType (section 3.1.4.1.1.9)</td>
<td>Specifies the websites for the contact. &lt;20&gt;</td>
</tr>
<tr>
<td>AbchEmailAddresses</td>
<td>t:AbchEmailAddressDictionaryType (section 3.1.4.1.1.2)</td>
<td>Represents an email address entry. This element is not present in server response.</td>
</tr>
<tr>
<td>Cid</td>
<td>xs:long</td>
<td>Specifies a unique handle to identify a user. &lt;21&gt;</td>
</tr>
<tr>
<td>SkypeAuthCertificate</td>
<td>xs:string</td>
<td>Specifies a token to connect to Skype. &lt;22&gt;</td>
</tr>
<tr>
<td>SkypeContext</td>
<td>xs:string</td>
<td>For internal use only. &lt;23&gt;</td>
</tr>
<tr>
<td>SkypeId</td>
<td>xs:string</td>
<td>The account name on Skype. &lt;24&gt;</td>
</tr>
<tr>
<td>SkypeRelationship</td>
<td>xs:string</td>
<td>For internal use only. &lt;25&gt;</td>
</tr>
<tr>
<td>YomiNickname</td>
<td>xs:string</td>
<td>Pronunciation guide for nickname field. &lt;26&gt;</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>XboxLiveTag</td>
<td>xs:string</td>
<td>Not used. &lt;27&gt;</td>
</tr>
<tr>
<td>InviteFree</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;28&gt;</td>
</tr>
<tr>
<td>HidePresenceAndProfile</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;29&gt;</td>
</tr>
<tr>
<td>IsPendingOutbound</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;30&gt;</td>
</tr>
<tr>
<td>SupportGroupFeeds</td>
<td>xs:boolean</td>
<td>Not used. &lt;31&gt;</td>
</tr>
<tr>
<td>UserTileHash</td>
<td>xs:string</td>
<td>Not used. &lt;32&gt;</td>
</tr>
<tr>
<td>UnifiedInbox</td>
<td>xs:boolean</td>
<td>For internal use only. &lt;33&gt;</td>
</tr>
<tr>
<td>Mris</td>
<td>t:ArrayOfStringsType</td>
<td>For internal use only. &lt;34&gt;</td>
</tr>
<tr>
<td>Wild</td>
<td>xs:string</td>
<td>For internal use only. &lt;35&gt;</td>
</tr>
<tr>
<td>AbchContactId</td>
<td>t:GuidType</td>
<td>For internal use only. &lt;36&gt;</td>
</tr>
<tr>
<td>NotInBirthdayCalendar</td>
<td>xs:boolean</td>
<td>Not used. &lt;37&gt;</td>
</tr>
<tr>
<td>DisplayNamePrefix</td>
<td>xs:string</td>
<td>Specifies the prefix for the display name. &lt;38&gt;</td>
</tr>
<tr>
<td>YomiGivenName</td>
<td>xs:string</td>
<td>Contains the name used in Japan for the searchable or phonetic spelling of a Japanese given name. &lt;39&gt;</td>
</tr>
<tr>
<td>YomiSurname</td>
<td>xs:string</td>
<td>Contains the name used in Japan for the searchable or phonetic spelling of a Japanese surname. &lt;40&gt;</td>
</tr>
<tr>
<td>PersonalNotes</td>
<td>xs:string</td>
<td>Specifies the user notes. &lt;41&gt;</td>
</tr>
<tr>
<td>PersonId</td>
<td>t:ItemIdType</td>
<td>Specifies the ID of the person. &lt;42&gt;</td>
</tr>
</tbody>
</table>

2.2.5 Simple Types

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

<table>
<thead>
<tr>
<th>Simple Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>t:ContactSourceType</td>
<td>Specifies whether a contact (2) or distribution list is located in the server database or in the directory service.</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveDirectory</td>
<td>Specifies that the contact (2) or distribution list is located in the directory service.</td>
</tr>
<tr>
<td>Store</td>
<td>Specifies that the contact (2) or distribution list is located in the server database.</td>
</tr>
</tbody>
</table>

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.

### Attributes

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

### Groups

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

### 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.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyItem</td>
<td>Defines a request to copy an item in a mailbox in the server.</td>
</tr>
<tr>
<td>CreateItem</td>
<td>Defines a request to create an item in the server.</td>
</tr>
<tr>
<td>DeleteItem</td>
<td>Defines a request to delete an item from a mailbox in the server.</td>
</tr>
<tr>
<td>GetItem</td>
<td>Defines a request to get an item from a mailbox in the server.</td>
</tr>
<tr>
<td>MoveItem</td>
<td>Defines a request to move an item in the server.</td>
</tr>
<tr>
<td>UpdateItem</td>
<td>Defines a request to update an item in a mailbox.</td>
</tr>
<tr>
<td>GetUserPhoto</td>
<td>Defines a request to retrieve the profile image for a mailbox.</td>
</tr>
<tr>
<td>SetUserPhoto</td>
<td>Defines a request to add a photo to a user’s account.</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Message Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:GetItemSoapIn</td>
<td>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 3.1.4.2.1 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.</td>
</tr>
</tbody>
</table>

Response

<table>
<thead>
<tr>
<th>Message Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:GetItemSoapOut</td>
<td>Specifies the SOAP message returned by the server in response.</td>
</tr>
</tbody>
</table>

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.

```xml
<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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>t:AbchEmailAddressTypeType (section 3.1.4.1.2.1)</td>
<td>Specifies the email type.</td>
</tr>
<tr>
<td>Address</td>
<td>xs:string (XMLSCHEMA2)</td>
<td>Specifies the email address.</td>
</tr>
<tr>
<td>IsMessengerEnabled</td>
<td>xs:boolean (XMLSCHEMA2)</td>
<td>Specifies whether this address is enabled for instant messaging.</td>
</tr>
<tr>
<td>Capabilities</td>
<td>xs:long (XMLSCHEMA2)</td>
<td>For internal use only.</td>
</tr>
</tbody>
</table>
3.1.4.1.1.2  t:AbchEmailAddressDictionaryType Complex Type

The AbchEmailAddressDictionaryType complex type specifies an email address entry. <44>

```xml
<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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>t:AbchEmailAddressDictionaryEntryType</td>
<td>An email address entry.</td>
</tr>
<tr>
<td></td>
<td>(section 3.1.4.1.1.2)</td>
<td></td>
</tr>
</tbody>
</table>

3.1.4.1.1.3  t:AbchPersonContactHandle Complex Type

The AbchPersonContactHandle complex type specifies the link between a person item and
contact. <45>

```xml
<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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceId</td>
<td>xs:string ([XMLSCHEMA2])</td>
<td>The source of the contact.</td>
</tr>
<tr>
<td>ObjectId</td>
<td>xs:string</td>
<td>Specifies the unique ID within the source.</td>
</tr>
<tr>
<td>AccountName</td>
<td>xs:string</td>
<td>Specifies the account name. Differentiator if more than 1 entry.</td>
</tr>
</tbody>
</table>

3.1.4.1.1.4  t:ArrayOfAbchPersonContactHandlesType Complex Type

The ArrayOfAbchPersonContactHandlesType complex type specifies a contact handle. <46>

```xml
<xs:complexType name="ArrayOfAbchPersonContactHandlesType">
    <xs:sequence>
    </xs:sequence>
</xs:complexType>
```
The following table lists the child element of the `ArrayOfAbchPersonContactHandlesType` complex type.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContactHandle</td>
<td>t:AbchPersonContactHandle</td>
<td>(section 3.1.4.1.1.3) The contact handle.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.1.5 t:CompleteNameType Complex Type

The `CompleteNameType` complex type represents the complete name of a contact.

```xml
<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

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

3.1.4.1.1.6 t:ContactsFolderType Complex Type

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

```xml
<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

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

### 3.1.4.1.1.7  t:ContactsViewType Complex Type

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

```xml
<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

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialName</td>
<td>xs:string [XMLSCHEMA2]</td>
<td>Contains the first name in a contacts list to return in a response.</td>
</tr>
<tr>
<td>FinalName</td>
<td>xs:string</td>
<td>Contains the last name in a contacts list to return in a response.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.1.8  t:ContactUrlDictionaryEntryType Complex Type
The `ContactUrlDictionaryEntryType` complex type represents a URL that is associated with a contact.<50>

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

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>t:ContactUrlKeyType</td>
<td>The Url type.</td>
</tr>
<tr>
<td>Address</td>
<td>xs:string</td>
<td>The Url.</td>
</tr>
<tr>
<td>Name</td>
<td>xs:string</td>
<td>What the url is used for.</td>
</tr>
</tbody>
</table>

3.1.4.1.1.9 `t:ContactUrlDictionaryType` Complex Type

The `ContactUrlDictionaryType` complex type specifies a Url associated with a contact <51>

```
<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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Url</td>
<td>t:ContactUrlDictionaryEntryType</td>
<td>A Url associated with a contact.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>t:EmailAddressKeyType (section 3.1.4.1.2.3)</td>
<td>Contains a value that identifies an e-mail address that is associated with a contact (2).</td>
</tr>
<tr>
<td>Name&lt;52&gt;</td>
<td>xs:string [XMLSCHEMA2]</td>
<td>Contains the display name associated with an e-mail address of the contact (2).</td>
</tr>
<tr>
<td>RoutingType&lt;53&gt;</td>
<td>xs:string</td>
<td>Contains the routing type associated with an e-mail address of the contact (2).</td>
</tr>
<tr>
<td>MailboxType&lt;54&gt;</td>
<td>t:MailboxTypeType (IMS-OXWSCDATA) section 2.2.5.18</td>
<td>Contains the type of mailbox that is represented by the e-mail address of the contact (2).</td>
</tr>
</tbody>
</table>

### 3.1.4.1.11 t:EmailAddressDictionaryType Complex Type

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

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

Child Elements

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>t:EmailAddressDictionaryEntryType (section 3.1.4.1.1.10)</td>
<td>Represents an e-mail address that is associated with a contact (2).</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>t:ImAddressKeyType (section 3.1.4.1.2.5)</td>
<td>Represents the instant messaging addresses for a contact (2).</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>t:ImAddressDictionaryEntryType (section 3.1.4.1.1.14)</td>
<td>Represents a collection of instant messaging addresses for a contact (2).</td>
</tr>
</tbody>
</table>

3.1.4.1.1.14  t:PhoneNumberDictionaryEntryType Complex Type

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

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>t:PhoneNumberKeyType</td>
<td>Represents types of telephone numbers for a contact (2).</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td>t:PhoneNumberDictionaryEntryType</td>
<td>Contains a telephone number for a contact (2).</td>
</tr>
</tbody>
</table>

### 3.1.4.1.16  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" />
  </xs:sequence>
</xs:complexType>
```
<xs:complexType name="t:PhysicalAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry" type="t:PhysicalAddressDictionaryEntryType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

### Child Elements

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Street</td>
<td>xs:string [XMLSCHEMA2]</td>
<td>Contains the street address for a contact (2) item.</td>
</tr>
<tr>
<td>City</td>
<td>xs:string</td>
<td>Contains the city name for a contact (2) item.</td>
</tr>
<tr>
<td>State</td>
<td>xs:string</td>
<td>Contains the state for a contact (2) item.</td>
</tr>
<tr>
<td>CountryOrRegion</td>
<td>xs:string</td>
<td>Contains the country or region for a contact (2) item.</td>
</tr>
<tr>
<td>PostalCode</td>
<td>xs:string</td>
<td>Contains the postal code for a contact (2) item.</td>
</tr>
</tbody>
</table>

### Attributes

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>t:PhysicalAddressKeyType (section 3.1.4.1.2.8)</td>
<td>Identifies the types of physical addresses for a contact (2).</td>
</tr>
</tbody>
</table>

### 3.1.4.1.17 t:PhysicalAddressDictionaryType Complex Type

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

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry</td>
<td><code>t:PhysicalAddressDictionaryEntryType</code> (section 3.1.4.1.1.16)</td>
<td>Contains information that defines a physical address, such as a street address.</td>
</tr>
</tbody>
</table>

### 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.<55>

```xml
<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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal</td>
<td>Personal email.</td>
</tr>
<tr>
<td>Business</td>
<td>Business email.</td>
</tr>
<tr>
<td>Other</td>
<td>Some other type.</td>
</tr>
<tr>
<td>Passport</td>
<td>Microsoft email account.</td>
</tr>
</tbody>
</table>

#### 3.1.4.1.2.2 `t:ContactUrlKeyType` Simple Type

The `ContactUrlKeyType` simple type represents types of URLs for a contact.<56>

```xml
<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:restriction>
</xs:simpleType>
```
The following table lists the values of the ContactUrlKeyType simple type.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal</td>
<td>A personal Url.</td>
</tr>
<tr>
<td>Business</td>
<td>A business Url.</td>
</tr>
<tr>
<td>Attachment</td>
<td>An attachment Url.</td>
</tr>
<tr>
<td>EbcDisplayDefinition</td>
<td>Not used</td>
</tr>
<tr>
<td>EbcFinalImage</td>
<td>Not used</td>
</tr>
<tr>
<td>EbcLogo</td>
<td>Not used</td>
</tr>
<tr>
<td>Feed</td>
<td>Url for a news feed.</td>
</tr>
<tr>
<td>Image</td>
<td>Url to the contact image.</td>
</tr>
<tr>
<td>InternalMarker</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>Other</td>
<td>For anything else.</td>
</tr>
</tbody>
</table>

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).

```xml
<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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmailAddress1</td>
<td>Identifies the first e-mail address for the contact (2).</td>
</tr>
<tr>
<td>EmailAddress2</td>
<td>Identifies the second e-mail address for the contact (2).</td>
</tr>
<tr>
<td>EmailAddress3</td>
<td>Identifies the third e-mail address for the contact (2).</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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:restriction>
</xs:simpleType>
```
<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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Indicates that the <strong>FileAs</strong> value is not constructed from properties of other contacts (2), but is represented by a string, saved &quot;as is&quot;.</td>
</tr>
<tr>
<td>LastCommaFirst</td>
<td>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.</td>
</tr>
<tr>
<td>FirstSpaceLast</td>
<td>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).</td>
</tr>
<tr>
<td>Company</td>
<td>Indicates that the company name is displayed.</td>
</tr>
<tr>
<td>LastCommaFirstCompany</td>
<td>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.</td>
</tr>
<tr>
<td>CompanyLastFirst</td>
<td>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.</td>
</tr>
<tr>
<td>LastFirst</td>
<td>Indicates that the contact (2) is displayed as the last name followed by the first name, a space, and the middle name.</td>
</tr>
<tr>
<td>LastFirstCompany</td>
<td>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.</td>
</tr>
<tr>
<td>CompanyLastCommaFirst</td>
<td>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.</td>
</tr>
<tr>
<td>LastFirstSuffix</td>
<td>Indicates that the contact (2) is displayed as the last name, the first name, a space, and the suffix for the contact.</td>
</tr>
<tr>
<td>LastSpaceFirstCompany</td>
<td>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.</td>
</tr>
<tr>
<td>CompanyLastSpaceFirst</td>
<td>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.</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LastSpaceFirst</td>
<td>Indicates that the contact (2) is displayed as the last name, followed by a space, the first name, and the middle name.</td>
</tr>
<tr>
<td>DisplayName&lt;57&gt;</td>
<td>Indicates that the contact (2) is displayed as the display name.</td>
</tr>
<tr>
<td>FirstName&lt;58&gt;</td>
<td>Indicates that the contact (2) is displayed as the first name.</td>
</tr>
<tr>
<td>LastFirstMiddleSuffix&lt;59&gt;</td>
<td>Indicates that the contact (2) is displayed as the last name, the first name, the middle name, and the suffix for the contact.</td>
</tr>
<tr>
<td>LastName&lt;60&gt;</td>
<td>Indicates that the contact (2) is displayed as the last name.</td>
</tr>
<tr>
<td>Empty&lt;61&gt;</td>
<td>Indicates that the contact (2) is displayed as empty.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.2.5 t:ImAddressKeyType Simple Type

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

```xml
<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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImAddress1</td>
<td>Identifies the first instant messaging address for the user.</td>
</tr>
<tr>
<td>ImAddress2</td>
<td>Identifies the second instant messaging address for the user.</td>
</tr>
<tr>
<td>ImAddress3</td>
<td>Identifies the third instant messaging address for the user.</td>
</tr>
</tbody>
</table>

### 3.1.4.1.2.6 t:PhoneNumberKeyType Simple Type

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

```xml
<xs:simpleType name="PhoneNumberKeyType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="PhoneNumber1" />
        <xs:enumeration value="PhoneNumber2" />
        <xs:enumeration value="PhoneNumber3" />
    </xs:restriction>
</xs:simpleType>
```
<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>
### Enumeration

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssistantPhone</td>
<td>Identifies the telephone number as the assistant's telephone number.</td>
</tr>
<tr>
<td>BusinessFax</td>
<td>Identifies the telephone number as a business fax number.</td>
</tr>
<tr>
<td>BusinessPhone</td>
<td>Identifies the telephone number as a business telephone number.</td>
</tr>
<tr>
<td>BusinessPhone2</td>
<td>Identifies the telephone number as a second business telephone number.</td>
</tr>
<tr>
<td>Callback</td>
<td>Identifies the telephone number as a callback number.</td>
</tr>
<tr>
<td>CarPhone</td>
<td>Identifies the telephone number as a car telephone number.</td>
</tr>
<tr>
<td>CompanyMainPhone</td>
<td>Identifies the telephone number as the company's main telephone number.</td>
</tr>
<tr>
<td>HomeFax</td>
<td>Identifies the telephone number as a home fax number.</td>
</tr>
<tr>
<td>HomePhone</td>
<td>Identifies the telephone number as a home telephone number.</td>
</tr>
<tr>
<td>HomePhone2</td>
<td>Identifies the telephone number as a second home telephone number.</td>
</tr>
<tr>
<td>Isdn</td>
<td>Identifies the telephone number as an Integrated Services Digital Network (ISDN) line.</td>
</tr>
<tr>
<td>MobilePhone</td>
<td>Identifies the telephone number as a mobile phone number.</td>
</tr>
<tr>
<td>OtherFax</td>
<td>Identifies the telephone number as another fax number.</td>
</tr>
<tr>
<td>OtherTelephone</td>
<td>Identifies the telephone number as another telephone number.</td>
</tr>
<tr>
<td>Pager</td>
<td>Identifies the telephone number as a pager.</td>
</tr>
<tr>
<td>PrimaryPhone</td>
<td>Identifies the telephone number as the primary telephone number.</td>
</tr>
<tr>
<td>RadioPhone</td>
<td>Identifies the telephone number as a radio telephone.</td>
</tr>
<tr>
<td>Telex</td>
<td>Identifies the telephone number as a telex telephone number.</td>
</tr>
<tr>
<td>TtyTddPhone</td>
<td>Identifies the telephone number as a teletype/telecommunication device for the deaf (TTY/TDD) telephone number.</td>
</tr>
<tr>
<td>BusinessMobile</td>
<td>Identifies the telephone number as a business mobile phone number.</td>
</tr>
<tr>
<td>IPPhone</td>
<td>Voice over IP phone number.</td>
</tr>
<tr>
<td>Mms</td>
<td>Not used.</td>
</tr>
<tr>
<td>Msn</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

### 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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Indicates that no type is specified for the address.</td>
</tr>
<tr>
<td>Business</td>
<td>Displays the address as a business address.</td>
</tr>
<tr>
<td>Home</td>
<td>Displays the address as a home address.</td>
</tr>
<tr>
<td>Other</td>
<td>Displays the address as an address of type other.</td>
</tr>
</tbody>
</table>

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.

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

**Request**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:DeleteItemSoapIn ([MS-OXWScore] section 3.1.4.3.1.1)</td>
<td>Specifies the SOAP message that defines the contact (2) item to delete. The <strong>DeleteItem</strong> ([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.</td>
</tr>
</tbody>
</table>

**Response**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:DeleteItemSoapOut ([MS-OXWScore] section 3.1.4.3.1.2)</td>
<td>Specifies the SOAP message returned by the server in response.</td>
</tr>
</tbody>
</table>

#### 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.

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

**Request**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:UpdateItemSoapIn ([MS-OXWScore] section 3.1.4.9.1.1)</td>
<td>Specifies the SOAP message that defines the contact (2) item to update. The <strong>Items</strong> child element of the <strong>UpdateItem</strong> ([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.</td>
</tr>
</tbody>
</table>
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.

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

**Request**

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

**Response**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:MoveItemSoapOut ([MS-OXWSCORE] section 3.1.4.7.1.2)</td>
<td>Specifies the SOAP message returned by the server in response.</td>
</tr>
</tbody>
</table>

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.

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

**Request**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:CopyItemSoapIn ([MS-OXWSCORE] section 3.1.4.1.2)</td>
<td>Specifies the SOAP message that defines the copy contact (2) item elements.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tns:CopyItemSoapOut ([MS-OXWSCORE] section 3.1.4.1.2)</td>
<td>Specifies the SOAP message returned by the server in response.</td>
</tr>
</tbody>
</table>
### 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.

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

**Request**

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

**Response**

<table>
<thead>
<tr>
<th>Message format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tns:CreateItemSoapOut</code> ([MS-OXWSCORE] section 3.1.4.2.1.2)</td>
<td>Specifies the SOAP message returned by the server in response.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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.

```xml
<wSDL:operation name="GetUserPhoto">
    <wsdl:input>
        <soap:header message="tns:GetUserPhotoSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" 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.

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUserPhotoSoapIn</td>
<td>Specifies a request to retrieve a photo.</td>
</tr>
<tr>
<td>GetUserPhotoSoapOut</td>
<td>Specifies the response to the <strong>GetUserPhotoSoapIn</strong> request <strong>WSDL message</strong></td>
</tr>
</tbody>
</table>

#### 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.

```xml
<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.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>tns:GetUserPhoto (section 3.1.4.7.2.1)</td>
<td>Specifies the <strong>SOAP body</strong> of the request to retrieve a photo.</td>
</tr>
<tr>
<td>RequestVersion</td>
<td>t:RequestServerVersion ([MS-OXWSCDATA] section)</td>
<td>Specifies a <strong>SOAP header</strong> that identifies the schema version for the <strong>GetUserPhoto</strong> <strong>WSDL operation</strong> request.</td>
</tr>
</tbody>
</table>
### 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.

```xml
<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.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUserPhotoResult</td>
<td>tns:GetUserPhotoResponse (section 3.1.4.7.2.2)</td>
<td>Specifies the SOAP body of the response to a <strong>GetUserPhoto</strong> WSDL operation request.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUserPhoto</td>
<td>Specifies the input data for the <strong>GetUserPhoto</strong> WSDL operation.</td>
</tr>
<tr>
<td>GetUserPhotoResponse</td>
<td>Specifies the result data for the <strong>GetUserPhoto</strong> WSDL operation.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<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.

```xml
<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.

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUserPhotoType</td>
<td>Specifies a request to retrieve a user photo.</td>
</tr>
<tr>
<td>GetUserPhotoResponseMessageType</td>
<td>Specifies the response message for the GetUserPhoto WSDL operation.</td>
</tr>
</tbody>
</table>

#### 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.

```xml
<x:s: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.

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>xs:string ([XMLSCHEMA2])</td>
<td>Specifies an email address.</td>
</tr>
<tr>
<td>SizeRequested</td>
<td>t:UserPhotoSizeType (section)</td>
<td>Specifies the requested size of the</td>
</tr>
</tbody>
</table>
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-OXWSDATA] section 2.2.4.67.

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

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

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HasChanged</td>
<td>xs:boolean ([XMLSCHEMA2])</td>
<td>Specifies whether the photo has changed. True, if the photo has changed.</td>
</tr>
<tr>
<td>PictureData</td>
<td>xs:base64Binary ([XMLSCHEMA2])</td>
<td>Specifies the binary data for the picture.</td>
</tr>
<tr>
<td>ContentType</td>
<td>xs:string ([XMLSCHEMA2])</td>
<td>Specifies the content (MIME) type of the photo. &lt;66&gt;</td>
</tr>
</tbody>
</table>

### 3.1.4.7.4 Simple Types

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

<table>
<thead>
<tr>
<th>Simple type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserPhotoSizeType</td>
<td>Specifies the size of the image.</td>
</tr>
</tbody>
</table>
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.

```xml
<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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR48x48</td>
<td>The image is 48 pixels high and 48 pixels wide.</td>
</tr>
<tr>
<td>HR64x64</td>
<td>The image is 64 pixels high and 64 pixels wide.</td>
</tr>
<tr>
<td>HR96x96</td>
<td>The image is 96 pixels high and 96 pixels wide.</td>
</tr>
<tr>
<td>HR120x120</td>
<td>The image is 120 pixels high and 120 pixels wide.</td>
</tr>
<tr>
<td>HR240x240</td>
<td>The image is 240 pixels high and 240 pixels wide.</td>
</tr>
<tr>
<td>HR360x360</td>
<td>The image is 360 pixels high and 360 pixels wide.</td>
</tr>
<tr>
<td>HR432x432</td>
<td>The image is 432 pixels high and 432 pixels wide.</td>
</tr>
<tr>
<td>HR504x504</td>
<td>The image is 504 pixels high and 504 pixels wide.</td>
</tr>
<tr>
<td>HR648x648</td>
<td>The image is 648 pixels high and 648 pixels wide.</td>
</tr>
</tbody>
</table>

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. <67>
The following is the **WSDL port type** specification of the **SetUserPhoto** WSDL operation.

```xml
<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.

```xml
<wsdl:operation name="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.

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetUserPhotoSoapIn</td>
<td>Specifies a request to set a photo in a mailbox.</td>
</tr>
<tr>
<td>SetUserPhotoSoapOut</td>
<td>Specifies the response to the SetUserPhotoSoapIn request WSDL message.</td>
</tr>
</tbody>
</table>

#### 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.

```xml
<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**

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

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>tns:SetUserPhoto</td>
<td>Specifies the <strong>SOAP body</strong> of the request to set a</td>
</tr>
</tbody>
</table>
### 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**.

```xml
<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.

<table>
<thead>
<tr>
<th>Part name</th>
<th>Element/type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetUserPhotoResult</td>
<td>tns:SetUserPhotoResponse (section 3.1.4.8.2.2)</td>
<td>Specifies the SOAP body of the response to a <strong>SetUserPhoto WSDL operation</strong> request.</td>
</tr>
<tr>
<td>ServerVersion</td>
<td>t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)</td>
<td>Specifies a SOAP header that identifies the server version for the response.</td>
</tr>
</tbody>
</table>

### 3.1.4.8.2 Elements

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetUserPhoto</td>
<td>Specifies the input data for the <strong>SetUserPhoto WSDL operation</strong>.</td>
</tr>
<tr>
<td>SetUserPhotoResponse</td>
<td>Specifies the result data for the <strong>SetUserPhoto WSDL operation</strong>.</td>
</tr>
</tbody>
</table>

### 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**.
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.

```xml
<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.

<table>
<thead>
<tr>
<th>Complex type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetUserPhotoType</td>
<td>Specifies a request to add a photo to a user account.</td>
</tr>
<tr>
<td>SetUserPhotoResponseMessageType</td>
<td>Specifies the response message for the <strong>SetUserPhoto</strong> WSDL operation.</td>
</tr>
<tr>
<td></td>
<td>This complex type extends the <strong>ResponseMessageType</strong> complex type, as specified by [MS-OXWSCDATA] section 2.2.4.67.</td>
</tr>
</tbody>
</table>

### 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.

```xml
<x:simpleType 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:simpleType>
```

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

<table>
<thead>
<tr>
<th>Element name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>t:NonEmptyStringType ([MS-OXWSCDATA] section 2.2.5.20)</td>
<td>Specifies an email address.</td>
</tr>
<tr>
<td>Content</td>
<td>xs:string</td>
<td>Specifies</td>
</tr>
<tr>
<td>Element name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="XMLSCHEMA2" /></td>
<td>the photo content.</td>
</tr>
</tbody>
</table>

### 3.1.4.8.3.2 SetUserPhotoResponseMessageType

The `SetUserPhotoResponseMessageType` complex type specifies the response message for the `SetUserPhoto` WSDL operation. This type extends the `ResponseMessageType` complex type, as specified by [MS-OXWSCDATA] section 2.2.4.67.

The following is the `SetUserPhotoResponseMessageType` complex type specification.

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

```xml
<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="AQMkADc4ZTU0Y2E4LWE5MjctNDNjO2l1M2YqLTExMzE4NzhhZjRlOQBBdQsX2N8VgcA1ouDJnFVkk2sDP7jPV6sQQAAAgEFAAAA1ouDjNFVkk2sDP7jPV6sQQAAAgk1AAAA"/>
      </ItemIds>
    </GetItem>
  </soap:Body>
</soap:Envelope>
```

4.2 Get a User Photo

The following example of the GetUserPhoto operation, as described in section 3.1.4.7.2.1, 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
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:s="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
<?xml version="1.0" encoding="utf-8"?>
<envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <header/>
  <response MajorVersion="15" MinorVersion="0" MajorBuildNumber="545"/>
</envelope>
```
<GetUserPhotoResponse ResponseClass="Success"
   xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
   <ResponseCode>NoError</ResponseCode>
   <HasChanged>true</HasChanged>
   <PictureData>/9jDBkSEw8UHRofHh0aHBwgJC4</PictureData>
</GetUserPhotoResponse>
5 Security

5.1 Security Considerations for Implementers
None.

5.2 Index of Security Parameters
None.
### 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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCONT.wsdl</td>
<td>Contains the WSDL for the implementation of this protocol.</td>
<td>6</td>
</tr>
<tr>
<td>MS-OXWSCONT-messages.xsd</td>
<td>Contains the XML schema message definitions that are used in this protocol.</td>
<td>section 7.1</td>
</tr>
<tr>
<td>MS-OXWSCONT-types.xsd</td>
<td>Contains the XML schema type definitions that are used in this protocol.</td>
<td>section 7.2</td>
</tr>
<tr>
<td>MS-OXWSCORE-messages.xsd</td>
<td>Contains XML schema message definitions that are referred to by this protocol.</td>
<td>[MS-OXWSCORE] section 7.1</td>
</tr>
</tbody>
</table>

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 or the MS-OXWSCONT-messages.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
<?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: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-OXWSCONT-messages.xsd"/>
      <xs:include schemaLocation="MS-OXWSCONT-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">
```

[MS-OXWSCONT] - v20191204
Contacts Web Service Protocol
Copyright © 2019 Microsoft Corporation
Release: December 4, 2019
<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
<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
<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"/>
</wsdl:output>
</wsdl:operation>
<soap:header message="tns:DeleteItemSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>

<wsdl:operation name="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: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">
<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">
<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">
<wsdl:input>
<soap:header message="tns:GetUserPhotoSoapIn" part="RequestVersion" use="literal"/>
<soap:body parts="request" use="literal"/>
</wsdl:input>
<wsdl:output>
</wsdl:output>
</wsdl:operation>

<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
    <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: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.

<table>
<thead>
<tr>
<th>Schema name</th>
<th>Prefix</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages schema</td>
<td>m</td>
<td>section 7.1</td>
</tr>
<tr>
<td>Types schema</td>
<td>t</td>
<td>section 7.2</td>
</tr>
</tbody>
</table>

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

7.1 Messages Schema

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

MS-OXWSCONT-messages.xsd includes the file listed 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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Defining specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCDATA-messages.xsd</td>
<td>[MS-OXWSCDATA] section 7.1</td>
</tr>
<tr>
<td>MS-OXWSCONT-types.xsd</td>
<td>section 7.2</td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="utf-8"?>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <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:element name="ContentType" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```
7.2 Types Schema

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.

<table>
<thead>
<tr>
<th>File name</th>
<th>Defining specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-OXWSCDATA-types.xsd</td>
<td>[MS-OXWSCDATA] section 7.2</td>
</tr>
</tbody>
</table>

```xml
<xs:schema
    xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
    elementFormDefault="qualified" version="Exchange2016" id="types">
  <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:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```
<xs:element name="SupportGroupFeeds" type="xs:boolean" minOccurs="0"/>
<xs:element name="UserNameTileHash" 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="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:element name="DisplayNamePrefix" type="xs:string" minOccurs="0"/>
<xs:element name="YomiGivenName" type="xs:string" minOccurs="0"/>
<xs:element name="YomiSurname" type="xs:string" minOccurs="0"/>
<xs:element name="PersonalNotes" type="xs:string" minOccurs="0"/>
<xs:element name="PersonId" type="t:ItemIdType" minOccurs="0"/>
<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="Address" type="xs:string" minOccurs="0"/>
    <xs:element name="Name" 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:simpleContent>
</xs:complexType>

<xs:complexType name="EmailAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Url" type="t:EmailAddressDictionaryEntryType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<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:simpleType>
<xs:complexType name="EmailAddressKeyType">
<xs:restriction base="xs:string">
<xs:enumeration value="EmailAddress1"/>
<xs:enumeration value="EmailAddress2"/>
<xs:enumeration value="EmailAddress3"/>
</xs:restriction>
</xs:simpleType>
<xs:complexType 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:complexType 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: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:restriction>
</xs:simpleType>
<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 updates to those products.

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019

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

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

<1> Section 2.2.4.1: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the AbchPersonItemType complex type.

<2> Section 2.2.4.1: The initial release of Exchange 2016 starts to include the PersonIdGuid element. Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016 CU6 and Exchange 2019 do not include this element.

<3> Section 2.2.4.1: Exchange 2007, Exchange 2010, and Exchange 2013 do not include the ExchangePersonIdGuid element.

<4> 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).

<5> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2010 SP1 do not include the following elements: PhoneticFirstName, PhoneticLastName, PhoneticFullName, Alias, Notes, Photo, UserSMIMECertificate, MSExchangeCertificate, DirectoryId, ManagerMailbox, and DirectReports. These elements were introduced in Exchange 2010 SP2.

<6> Section 2.2.4.3: Exchange 2007 does not support the HasPicture element.

<7> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the AccountName element.

<8> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the IsAutoUpdateDisabled element.

<9> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the Comment element.

<10> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the ContactShortId element.

<11> Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the ContactType element.
Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the Gender element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the IsHidden element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the ObjectId element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the PassportId element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the IsPrivate element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SourceId element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the TrustLevel element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the CreatedBy element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the URLs element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the Cid element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SkypeAuthCertificate element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SkypeContext element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SkypeId element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SkypeRelationship element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the YomiNickname element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the XboxLiveTag element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the InviteFree element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the HidePresenceAndProfile element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the IsPendingOutbound element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the SupportGroupFeeds element.
Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **UserTileHash** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **UnifiedInbox** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Mris** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Wlid** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchContactId** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **NotInBirthdayCalendar** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **DisplayNamePrefix** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **YomiGivenName** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **YomiSurname** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **PersonalNotes** element.

Section 2.2.4.3: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **PersonId** element.

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

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

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

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

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

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

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

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

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

Section 3.1.4.1.1.10: Exchange 2007 does not support the **Name** attribute.
<53> **Section 3.1.4.1.1.10**: Exchange 2007 does not support the **RoutingType** attribute.

<54> **Section 3.1.4.1.1.10**: Exchange 2007 does not support the **MailboxType** attribute.

<55> **Section 3.1.4.1.2.1**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **AbchEmailAddressTypeType** simple type.

<56> **Section 3.1.4.1.2.2**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **ContactUrlKeyType** simple type.

<57> **Section 3.1.4.1.2.4**: Exchange 2007 does not support the **DisplayName** attribute.

<58> **Section 3.1.4.1.2.4**: Exchange 2007 does not support the **FirstName** attribute.

<59> **Section 3.1.4.1.2.4**: Exchange 2007 does not support the **LastFirstMiddleSuffix** attribute.

<60> **Section 3.1.4.1.2.4**: Exchange 2007 does not support the **LastName** attribute.

<61> **Section 3.1.4.1.2.4**: Exchange 2007 does not support the **Empty** attribute.

<62> **Section 3.1.4.1.2.6**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **BusinessMobile** value.

<63> **Section 3.1.4.1.2.6**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **IPPhone** value.

<64> **Section 3.1.4.1.2.6**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Mms** value.

<65> **Section 3.1.4.1.2.6**: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Msn** value.

<66> **Section 3.1.4.7.3.2**: Exchange 2007 and Exchange 2010 do not support the **ContentType** element.

<67> **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 Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 None means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.4.1 t:AbchPersonItemType Complex Type</td>
<td>Updated schema for AbchPersonItemType complex type.</td>
<td>Minor</td>
</tr>
<tr>
<td>2.2.4.1 t:AbchPersonItemType Complex Type</td>
<td>Added a product behavior note for PersonIdGuid element.</td>
<td>Minor</td>
</tr>
<tr>
<td>2.2.4.3 t:ContactItemType Complex Type</td>
<td>Updated description for type of element PersonId.</td>
<td>Minor</td>
</tr>
<tr>
<td>2.2.4.3 t:ContactItemType Complex Type</td>
<td>Updated description for AbchEmailAddresses element.</td>
<td>Minor</td>
</tr>
<tr>
<td>3.1.4.1 1 t:AbchEmailAddressDictionaryEntryType Complex Type</td>
<td>Updated product behavior note for t:AbchEmailAddressDictionaryEntryType type.</td>
<td>Minor</td>
</tr>
<tr>
<td>3.1.4.7.3 Complex Types</td>
<td>Removed GetUserPhotoResponseType complex type.</td>
<td>Minor</td>
</tr>
<tr>
<td>7.1 Messages Schema</td>
<td>Removed schema for GetUserPhotoResponseType complex type.</td>
<td>Minor</td>
</tr>
<tr>
<td>7.2 Types Schema</td>
<td>Updated type schema for element PersonId in ContactItemType type.</td>
<td>Minor</td>
</tr>
<tr>
<td>7.2 Types Schema</td>
<td>Updated schema for AbchPersonItemType complex type.</td>
<td>Minor</td>
</tr>
</tbody>
</table>
CopyItem operation 45
CreateItem operation 46
DeleteItem operation 44
GetItem operation 25
GetUserPhoto operation 46
initialization 25
local events 55
message processing 25
MoveItem operation 45
sequencing rules 25
SetUserPhoto operation 51
timer events 55
timers 25
UpdateItem operation 44
Simple types 23
  t:ContactSourceType Simple Type 23
Standards assignments 11
Syntax
    messages - overview 12

T
  t:AbchPersonItemType Complex Type complex type 13
  t:ArrayOfBinaryType Complex Type complex type 14
  t:ContactItemType Complex Type complex type 14
  t:ContactSourceType Simple Type simple type 23
Timer events
  server 55
Timers
  server 25
Tracking changes 76
Transport 12
Types
  complex 13
  simple 23

V
Vendor-extensible fields 11
Versioning 10

W
WSDL 59

X
XML schema 64
  Messages Schema 64
  Types Schema 65