[MS-OXWSPHOTO]:
Photo Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/16/2012	0.1	New	Released new document.
10/08/2012	1.0	Major	Significantly changed the technical content.
02/11/2013	2.0	Major	Significantly changed the technical content.
07/26/2013	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	3.0	Major	Significantly changed the technical content.
07/31/2014	4.0	Major	Significantly changed the technical content.
10/30/2014	4.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Photo Web Service Protocol enables the transfer of a user photo from a **mailbox** to a client application that can authenticate and send an **HTTP GET** request.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

email address
endpoint
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
mailbox
SOAP
SOAP action
SOAP body
SOAP header
web service
Web Services Description Language (WSDL)
WSDL message
WSDL operation
WSDL port type
XML schema

The following terms are specific to this document:

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

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

1.2.1 Normative References

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

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

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

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

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

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

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

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

1.2.2 Informative References

[MS-OCAUTHWS] Microsoft Corporation, "OC Authentication Web Service Protocol".

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary".

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

1.3 Overview

The Photo Web Service Protocol enables client applications to use a **web service** to request an image that represents a mailbox. This image, typically a photo of the mailbox owner, can be used by a client application to identify the mailbox.

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].

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

1.5 Prerequisites/Preconditions

This protocol is accessible only to authenticated users, either directly through a client application or indirectly through a trusted server application. This protocol uses the OC Authentication Web Service Protocol, as described in [MS-OCAUTHWS], for authentication.

1.6 Applicability Statement

This protocol applies to environments that use a web service to transfer images.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2   Messages

2.1   Transport

This protocol is transported by **HTTPS**, as specified in [RFC2818].

2.2   Message Syntax

None.

3   Protocol Details

3.1   Server Details

This section applies to the REST **endpoint** (4) for this protocol.

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Higher-Layer Triggered Events

None.

3.1.5   Message Processing Events and Sequencing Rules

This protocol manipulates the resource listed in the following table.

Resource	Description
UserPhoto	The profile image for a mailbox.

The responses to all the operations can result in the status codes listed in the following table.

Status code	Description
200	An image is available for the specified mailbox, and the binary image is the contents of the response.
304	The image has not changed since the **ETag** header was returned to the client application.
400	The request could not be understood by the server due to malformed syntax.
401	The request requires user authentication.
404	No image is available for the specified mailbox.

The server returns an **ETag** header, as specified in [RFC2616], in the response to the request for a user image. The **ETag** header remains the same for the user image until the image is updated. You can return this **ETag** header to the server in the **HTTPS GET** request for the user image in an **If-None-Match** header, as specified in [RFC2616]. If the image has not changed since the last request, the server responds with an HTTP 304 response that indicates that the image has not changed since the last request.

3.1.5.1   UserPhoto

The following table lists the operations that are allowed to be performed on this resource.

Operation	Description
GetUserPhoto	Retrieves the profile image for a mailbox.

3.1.5.1.1   GetUserPhoto

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

```
https://<Exchange Server>/ews/Exchange.asmx/s/GetUserPhoto?email=<email address>&size=<size
code>
```

The Autodiscover service **GetUserSetting WSDL operation**, as specified in [MS-OXWSADISC], is used to retrieve the **ExternalPhotosUrl** setting, which contains the URL of the web service endpoint (4) and the location of the Exchange.asmx **HTTP** handler that returns the user images.

email: Represents the **email address** of the user account.

size: Contains the size code of the user image. The following table describes possible values. The size code always returns the directory service thumbnail image if it is available as long as no image is stored on the server.

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

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.

The **Accept** header, as specified in [RFC2616], is not processed by the server.

Response:

The requested image is returned in the payload of the HTTP response. The type of the image is indicated by the **Content-Type** header, as specified in [RFC2616]. Optionally, the **ETag** header, as specified in [RFC2616], is also returned.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 ExchangeServicePortType Server Details

This section applies to the **SOAP** endpoint (4) for this protocol.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

For the SOAP endpoint (4), this protocol defines a single **WSDL port type** with one operation that allows users to retrieve a profile image.

The following table summarizes the operation defined by this specification.

Operation	Description
GetUserPhoto	Retrieves a profile image for a mailbox.

3.2.5.1 GetUserPhoto

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

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

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

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

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

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

3.2.5.1.1   Messages

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

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

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

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

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

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

Part name	Element/type	Description
request	**tns:GetUserPhoto**	Specifies the **SOAP body** of the request to retrieve a photo.
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.9)	Specifies a **SOAP header** that identifies the schema version for the **GetUserPhoto** WSDL operation request.

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

```
<wsdl:message name="GetUserPhotoSoapOut">
```

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

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

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

Part name	Element/type	Description
GetUserPhotoResult	**tns:GetUserPhotoResponse**	Specifies the SOAP body of the response to a **GetUserPhoto** WSDL operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.10)	Specifies a SOAP header that identifies the server version for the response.

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

If the **GetUserPhoto** WSDL operation is not successful, it returns a **GetUserPhotoResponse**
element with the **ResponseClass** attribute set to "Error". The **ResponseCode** element of the
GetUserPhotoResponse element is set to one of the common errors defined in [MS-OXWSCDATA]
section 2.2.5.23.

3.2.5.1.2   Elements

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

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

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

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

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

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

.

3.2.5.1.3 Complex Types

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

Complex type	Description
GetUserPhotoType	Specifies a request to retrieve a user photo. This complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.15.
GetUserPhotoResponseMessageType	Specifies the response message for the **GetUserPhoto** WSDL operation. This complex type extends the **ResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.57.
GetUserPhotoResponseType	Specifies the response for the **GetUserPhoto** WSDL operation. This complex type extends the **BaseResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.16.

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

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

Email: An element of type **string** type, as defined in [XMLSCHEMA2], that specifies an email address.

SizeRequested: An element of type **UserPhotoSizeType** that specifies the requested size of the photo.

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

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

```
<xs:complexType name="GetUserPhotoResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="HasChanged" type="xs:boolean"
              minOccurs="1" maxOccurs="1"/>
```

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

HasChanged: An element of type **xs:boolean**, as defined in [XMLSCHEMA2], that specifies whether the photo has changed.

PictureData: An element of type **xs:base64Binary**, as defined in [XMLSCHEMA2], that specifies the binary data for the picture.

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

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

3.2.5.1.4 Simple Types

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

Simple type	Description
UserPhotoSizeType	Specifies the size of the image.

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

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

```
  <xs:simpleType name="UserPhotoSizeType">
     <xs:restriction base="xs:string">
       <xs:enumeration value="HR48x48" />
       <xs:enumeration value="HR64x64" />
       <xs:enumeration value="HR96x96" />
       <xs:enumeration value="HR120x120" />
       <xs:enumeration value="HR240x240" />
       <xs:enumeration value="HR360x360" />
       <xs:enumeration value="HR432x432" />
       <xs:enumeration value="HR504x504" />
       <xs:enumeration value="HR648x648" />
     </xs:restriction>
  </xs:simpleType>
```

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

Value	Meaning
HR48x48	The image is 48 pixels high and 48 pixels wide.

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

3.2.5.1.5 Attributes

None.

3.2.5.1.6 Groups

None.

3.2.5.1.7 Attribute Groups

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4   Protocol Examples

The following example of the **GetUserPhoto** operation, as described in section 3.2.5.1, shows how the client retrieves a photo by using SOAP. This example requests a photo 96 pixels high and 96 pixels wide associated with the email address "user1@contoso.com".

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

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

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="15"
             MinorVersion="0"
             MajorBuildNumber="545"
             MinorBuildNumber="5"
             Version="Exchange2013"
             xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
             xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <GetUserPhotoResponse ResponseClass="Success"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <ResponseCode>NoError</ResponseCode>
      <HasChanged>true</HasChanged>
      <PictureData>/9jDBkSEw8UHRofHh0aHBwgJC4</PictureData>
    </GetUserPhotoResponse>
  </s:Body>
</s:Envelope>
```

Example using REST interface

Request (HTTP GET)

```
https://outlook.office365.com/ews/Exchange.asmx/s/GetUserPhoto?email=user1%40contoso.com&size
=HR120x120
```

Response

Headers

Content-Typeimage/jpeg

ETag"889B7442"

Body (payload)

5   Security

5.1   Security Considerations for Implementers

This protocol relies on the web server that hosts the application to perform authentication.

5.2   Index of Security Parameters

None.

6   Appendix A: Full XML Schema

The following table lists the XML files that are required to implement the functionality that is specified in this document. The contents of each file are included in this section.

File name	Description	Section
MS-OXWSPHOTO.wsdl	Contains the WSDL for the implementation of this protocol.	6.1
MS-OXWSPHOTO-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	6.2
MS-OXWSPHOTO-types.xsd	Contains the XML schema type definitions that are used in this protocol.	6.3

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-OXWSPHOTO-types.xsd or MS-OXWSPHOTO-messages.xsd schema have to be placed in the common folder along with the files.

6.1   WSDL Schema

This section contains the contents of the MS-OXWSPHOTO.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      targetNamespace=
            "http://schemas.microsoft.com/exchange/services/2006/messages">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2013"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          targetNamespace=
                "http://schemas.microsoft.com/exchange/services/2006/messages">
      <xs:include schemaLocation="MS-OXWSPHOTO-messages.xsd"/>
    </xs:schema>
  </wsdl:types>
  <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:portType name="ExchangeServicePortType">
    <wsdl:operation name="GetUserPhoto">
      <wsdl:input message="tns:GetUserPhotoSoapIn"/>
      <wsdl:output message="tns:GetUserPhotoSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <soap:binding style="document"
          transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:documentation>
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0"
```

```
              xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/"/>
      </wsdl:documentation>
      <wsdl:operation name="GetUserPhoto">
        <soap:operation soapAction=
      "http://schemas.microsoft.com/exchange/services/2006/messages/GetUserPhoto"/>
        <wsdl:input>
          <soap:body parts="request" use="literal"/>
          <soap:header message="tns:GetUserPhotoSoapIn"
               part="RequestVersion" use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body parts="GetUserPhotoResult" use="literal"/>
          <soap:header message="tns:GetUserPhotoSoapOut"
               part="ServerVersion" use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```

6.2   Messages Schema

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

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

File name	Defining specification
MS-OXWSCDATA-messages.xsd	[MS-OXWSCDATA] section 7.1

```
  <?xml version="1.0" encoding="utf-8"?>
  <xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        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/messages"
            elementFormDefault="qualified"
            version="Exchange2013" id="messages">
    <xs:import namespace=
        "http://schemas.microsoft.com/exchange/services/2006/types"
        schemaLocation="MS-OXWSPHOTO-types.xsd"/>
    <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"/>
```

```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:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetUserPhotoResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="GetUserPhotoResponse"
      type="m:GetUserPhotoResponseMessageType"/>
</xs:schema>
```

6.3 Types Schema

This section contains the contents of the MS-OXWSPHOTO-types.xsd file.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="types"
      elementFormDefault="qualified"
      version="Exchange2013"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:simpleType name="UserPhotoSizeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="HR48x48" />
      <xs:enumeration value="HR64x64" />
      <xs:enumeration value="HR96x96" />
      <xs:enumeration value="HR120x120" />
      <xs:enumeration value="HR240x240" />
      <xs:enumeration value="HR360x360" />
      <xs:enumeration value="HR432x432" />
      <xs:enumeration value="HR504x504" />
      <xs:enumeration value="HR648x648" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

7   Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Exchange Server 2013

- Microsoft Lync 2013

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9   Index

T

Timer events
   ExchangeServicePortType server 15
   server 10
Timers
   ExchangeServicePortType server 10
   server 8
Tracking changes 23
Transport 7

V

Vendor-extensible fields 6
Versioning 6