[MS-OXWSATT]:

Attachment Handling Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Copyright © 2015 Microsoft Corporation

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.0.1	Editorial	Revised and edited the technical content.
2/10/2010	2.0.0	Major	Updated and revised the technical content.
5/5/2010	3.0.0	Major	Updated and revised the technical content.
8/4/2010	3.1	Minor	Clarified the meaning of the technical content.
11/3/2010	4.0	Major	Significantly changed the technical content.
3/18/2011	5.0	Major	Significantly changed the technical content.
8/5/2011	5.1	Minor	Clarified the meaning of the technical content.
10/7/2011	5.1	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	6.0	Major	Significantly changed the technical content.
4/27/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.0	Major	Significantly changed the technical content.
10/8/2012	7.1	Minor	Clarified the meaning of the technical content.
2/11/2013	7.1	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	7.2	Minor	Clarified the meaning of the technical content.
11/18/2013	8.0	Major	Significantly changed the technical content.
2/10/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	9.0	Major	Significantly changed the technical content.
7/31/2014	9.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	10.0	Major	Significantly changed the technical content.
5/26/2015	11.0	Major	Significantly changed the technical content.
9/14/2015	12.0	Major	Significantly changed the technical content.

Copyright © 2015 Microsoft Corporation

Table of Contents

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation

1 Introduction

The Attachment Handling Web Service Protocol is used to create, delete, and get attachments on items on the server.

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

1.1 Glossary

The following terms are specific to this document:

Attachment object: A set of properties that represents a file, **Message object**, or structured storage that is attached to a Message object and is visible through the attachments table for a Message object.

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].

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

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

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

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

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

Object Linking and Embedding (OLE): A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either embedded or linked. See also embedded object and linked object.

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 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.

Copyright © 2015 Microsoft Corporation

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 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 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.

Copyright © 2015 Microsoft Corporation

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

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

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

[MS-OXWSCONT] Microsoft Corporation, "Contacts Web Service Protocol".

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

[MS-OXWSGTZ] Microsoft Corporation, "Get Server Time Zone Web Service Protocol".

[MS-OXWSMSG] Microsoft Corporation, "Email Message Types Web Service Protocol".

[MS-OXWSMTGS] Microsoft Corporation, "Calendaring Web Service Protocol".

[MS-OXWSPOST] Microsoft Corporation, "Post Items Web Service Protocol".

[MS-OXWSTASK] Microsoft Corporation, "Tasks Web Service Protocol".

[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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001, http://www.ietf.org/rfc/rfc3066.txt

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

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

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

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

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

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

Copyright © 2015 Microsoft Corporation

1.2.2 Informative References

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

1.3 Overview

The Attachment Handling Web Service Protocol enables clients to create, get, and delete attachments on messages. Clients use **Attachment objects** to associate files, **Object Linking and Embedding (OLE)** objects, other messages, or binary data with a particular **Message object**. Because Attachment objects are created, maintained, and accessed only in the context of a message, they are considered to be subobjects.

1.4 Relationship to Other Protocols

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

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

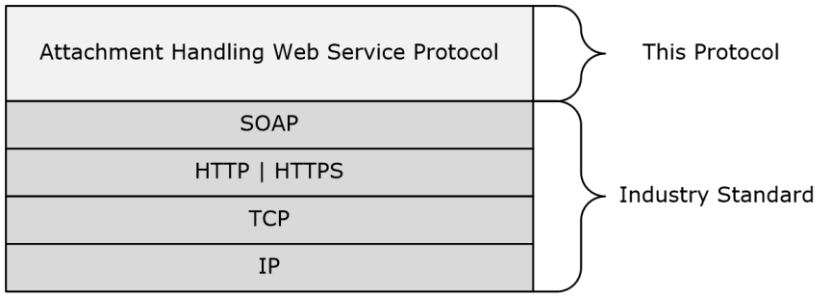

Figure 1: This protocol in relation to other protocols

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 and Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], 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

The protocol specified in this document is applicable to environments that create, get, and delete attachments on messages. This Web service protocol is applicable to SOAP-based clients [SOAP1.1].

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

Copyright © 2015 Microsoft Corporation

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section 2.1.
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **WSDL** version of the request is identified using the **t:RequestServerVersion** element as described in [MS-OXWSCDATA] section 2.2.3.11, and the version of the server responding to the request is identified using the **t:ServerVersionInfo** element as described in [MS-OXWSCDATA] section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in sections 2.2 and 3.1.4.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Copyright © 2015 Microsoft Corporation

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. For details, see [SOAP1.1].

This protocol relies on the Web server that hosts the application to perform authentication. This protocol MUST support SOAP over HTTP, as specified in [RFC2616]. The protocol SHOULD use secure communications via HTTPS, as defined 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 Web Services Description Language (WSDL), as defined in [WSDL].

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
wsi	http://ws-i.org/schemas/conformanceClaim/	[WSIBASIC]
t	http://schemas.microsoft.com/exchange/services/2006/types	
m	http://schemas.microsoft.com/exchange/services/2006/messages	

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.

Copyright © 2015 Microsoft Corporation

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 described with the operation.

Complex type	Description
ArrayOfAttachmentsType	Represents an array of types based on attachments on the item.
AttachmentIdType	Represents the item identifier and new change key of an item after an attachment has been attached.
AttachmentInfoResponseMessageType	Contains status and response data for attachments.
AttachmentType	Represents an attachment.
FileAttachmentType	Represents a file that is attached to an item in the server store.
ItemAttachmentType	Represents an item that is attached to another item.
NonEmptyArrayOfRequestAttachmentIdsType	Represents an array of attachment identifiers.
ReferenceAttachmentType	Represents a reference that is attached another item.
RequestAttachmentIdType	Represents the identifier for an attachment.

2.2.4.1 t:ArrayOfAttachmentsType Complex Type

The **ArrayOfAttachmentsType** complex type represents an array of types based on attachments on the item. The **ArrayOfAttachmentsType** complex type is used only in the response message.

```
<xs:complexType name="ArrayOfAttachmentsType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="ItemAttachment"
      type="t:ItemAttachmentType"
     />
    <xs:element name="FileAttachment"
      type="t:FileAttachmentType"
     />
   <xs:element name="ReferenceAttachment"
      type="t:ReferenceAttachmentType"/>
  </xs:choice>
</xs:complexType>
```

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

Element	Type	Description
ItemAttachment	**t:ItemAttachmentType** (section 2.2.4.6)	Specifies an item that is attached to another item.
FileAttachment	**t:FileAttachmentType** (section 2.2.4.5)	Specifies a file that is attached to

Copyright © 2015 Microsoft Corporation

Element	Type	Description
		another item.
ReferenceAttachment	**t:ReferenceAttachmentType** (section 2.2.4.8)	Specifies an reference that is attached to another item.

2.2.4.2 t:AttachmentIdType Complex Type

The **AttachmentIdType** complex type represents the item identifier and new change key of an item after an attachment has been attached. The **AttachmentIdType** complex type extends the **RequestAttachmentIdType** complex type, as specified in section 2.2.4.9.

```
<xs:complexType name="AttachmentIdType">
  <xs:complexContent>
    <xs:extension
      base="t:RequestAttachmentIdType"
    >
      <xs:attribute name="RootItemId"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="RootItemChangeKey"
        type="xs:string"
        use="optional"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table summarizes the set of common XML schema attribute definitions defined by this specification.

Attribute	Type	Description
RootItemId	**xs:string** ([XMLSCHEMA2])	Represents the unique identifier of the root store item to which the attachment is attached. The store item is an item in the server store. The maximum length is 512 bytes after **base64** decoding.
RootItemChangeKey	**xs:string**	Represents the change key of the root store item to which the attachment is attached. The maximum length is 512 bytes after base64 decoding.

2.2.4.3 m:AttachmentInfoResponseMessageType Complex Type

The **AttachmentInfoResponseMessageType** complex type contains status and response data for attachments. The **AttachmentInfoResponseMessageType** complex type extends the **ResponseMessageType** complex type, ([MS-OXWSCDATA] section 2.2.4.67).

```
<xs:complexType name="AttachmentInfoResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
```


Copyright © 2015 Microsoft Corporation

```
    >
      <xs:sequence>
        <xs:element name="Attachments"
          type="t:ArrayOfAttachmentsType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element	Type	Description
Attachments	**t:ArrayOfAttachmentsType** (section 2.2.4.1)	Represents an array of types based on attachments on the item. The **ArrayOfAttachmentsType** complex type is used only in the response message.

2.2.4.4 t:AttachmentType Complex Type

The **AttachmentType** complex type represents an attachment.

```
<xs:complexType name="AttachmentType">
  <xs:sequence>
    <xs:element name="AttachmentId"
      type="t:AttachmentIdType"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="Name"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="ContentType"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="ContentId"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="ContentLocation"
      type="xs:string"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="Size"
      type="xs:int"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="LastModifiedTime"
      type="xs:dateTime"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="IsInline"
```


Copyright © 2015 Microsoft Corporation

```
        type="xs:boolean"
        minOccurs="0"
        maxOccurs="1"
       />
    </xs:sequence>
  </xs:complexType>
```

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

Element	Type	Description
AttachmentId	**t:AttachmentIdType** (section 2.2.4.2)	Specifies the attachment identifier.
Name	**xs:string**([XMLSCHEMA2])	Specifies the descriptive name of the attachment.
ContentType	**xs:string**	Specifies the **MIME** type of the attachment content. For example: "text/enriched", "text/html", "text/plain", "text/rfc822-headers", "text/richtext", or "text/sgml".
ContentId	**xs:string**	Specifies the unique object identifier for an attachment.
ContentLocation	**xs:string**	Specifies the **URI** that corresponds to the location of the content of the attachment. The **ContentLocation** element can be used to associate an attachment with a URL that defines its location on the Web.
Size	**xs:int** ([XMLSCHEMA2])	Specifies an estimate of the size, in bytes, of the item's complete body.
LastModifiedTime	**xs:dateTime** ([XMLSCHEMA2])	Specifies the day and time that this item was last changed.
IsInline<1>	**xs:boolean** ([XMLSCHEMA2])	A Boolean value that indicates whether the attachment is an inline attachment. The **IsInline** element is set to indicate that the attachment appears inline within an item.

The **AttachmentType** complex type is extended by the **ItemAttachmentType** complex type, as specified in section 2.2.4.6, and the **FileAttachmentType** complex type, as specified in section 2.2.4.5.

2.2.4.5 t:FileAttachmentType Complex Type

The **FileAttachmentType** complex type represents a file that is attached to an item in the server store. The **FileAttachmentType** complex type extends the **AttachmentType** complex type, as specified in section 2.2.4.4.

```
<xs:complexType name="FileAttachmentType">
  <xs:complexContent>
    <xs:extension name="FileAttachmentType"
      base="t:AttachmentType"
    >
      <xs:sequence>
        <xs:element name="IsContactPhoto"
          type="xs:boolean"
          minOccurs="0"
          maxOccurs="1"
         />
        <xs:element name="Content"
          type="xs:base64Binary"
          minOccurs="0"
```


Copyright © 2015 Microsoft Corporation

```
            maxOccurs="1"
           />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

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

Element	Type	Description
IsContactPhoto<2>	**xs:boolean** ([XMLSCHEMA2])	A value that indicates whether this attachment is a contact photo. A text value of "true" indicates that the attachment is a contact photo.
Content	**xs:base64Binary** ([XMLSCHEMA2])	Represents the base64-encoded contents of the file attachment.

2.2.4.6 t:ItemAttachmentType Complex Type

The **ItemAttachmentType** complex type represents an item that is attached to another item in the server store. The **ItemAttachmentType** complex type extends the **AttachmentType** complex type as specified in section 2.2.4.4. If the item attachment is **MeetingMessage**, **MeetingRequest**, **MeetingResponse** or **MeetingCancellation**, the server SHOULD return **ErrorInvalidItemForOperationCreateItemAttachment** response code as specified in [MS-OXWSCDATA] section 2.2.5.24. If none of the child elements of **ItemAttachmentType** is specified in the **CreateAttachment** request, the server MUST return an **ErrorMissingItemForCreateItemAttachment** response code as specified in [MS-OXWSCDATA] section 2.2.5.24.

```
<xs:complexType name="ItemAttachmentType">
  <xs:complexContent>
    <xs:extension
      base="t:AttachmentType"
    >
      <xs:choice
        minOccurs="0"
        maxOccurs="1"
      >
        <xs:element name="Item"
          type="t:ItemType"
         />
        <xs:element name="Message"
          type="t:MessageType"
         />
        <xs:element name="CalendarItem"
          type="t:CalendarItemType"
         />
        <xs:element name="Contact"
          type="t:ContactItemType"
         />
        <xs:element name="MeetingMessage"
          type="t:MeetingMessageType"
         />
        <xs:element name="MeetingRequest"
          type="t:MeetingRequestMessageType"
         />
        <xs:element name="MeetingResponse"
          type="t:MeetingResponseMessageType"
```


Copyright © 2015 Microsoft Corporation

```
          />
          <xs:element name="MeetingCancellation"
            type="t:MeetingCancellationMessageType"
          />
          <xs:element name="Task"
            type="t:TaskType"
          />
          <xs:element name="PostItem"
            type="t:PostItemType"
          />
          <xs:element name="RoleMember"
            type="t:RoleMemberItemType"
          />
          <xs:element name="Network"
            type="t:NetworkItemType"
          />
          <xs:element name="Person"
            type="t:AbchPersonItemType"
          />
          <xs:element name="Booking"
            type="t:BookingItemType"
          />
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

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

Element	Type	Description
Item	**t:ItemType** ([MS-OXWSCORE] section 2.2.4.24)	Represents a generic item in the server store.<3>
Message	**t:MessageType** ([MS-OXWSMSG] section 2.2.4.1)	Represents a server e-mail message.
CalendarItem	**t:CalendarItemType** ([MS-OXWSMTGS] section 2.2.4.9)	Represents a calendar item.
Contact	**t:ContactItemType** ([MS-OXWSCONT] section 2.2.4.3)	Represents a contact item.
MeetingMessage	**t:MeetingMessageType** ([MS-OXWSMTGS] section 2.2.4.20)	Represents a meeting message in the server store.
MeetingRequest	**t:MeetingRequestMessageType** ([MS-OXWSMTGS] section 2.2.4.22)	Represents a meeting request in the server store.
MeetingResponse	**t:MeetingResponseMessageType** ([MS-OXWSMTGS] section 2.2.4.23)	Represents a meeting response in the server store.
MeetingCancellation	**t:MeetingCancellationMessageType** ([MS-OXWSMTGS] section 2.2.4.19)	Represents a meeting cancellation in the server store.
Task	**t:TaskType** ([MS-OXWSTASK] section 2.2.4.6)	Represents a task in the server store.
PostItem	**t:PostItemType** ([MS-OXWSPOST] section 2.2.4.1)	Represents a post item in the server store.
RoleMember	**t:RoleMemberItemType** ([MS-OXWSCORE] section	For internal use only.<4>

Copyright © 2015 Microsoft Corporation

Element	Type	Description
	2.2.4.42)	
Network	**t:NetworkItemType** ([MS-OXWSCORE] section 2.2.4.29)	For internal use only.<5>
Person	**t:AbchPersonItemType** ([MS-OXWSCONT] section 2.2.4.1)	Represents a person in the server store.<6>
Booking	**t:BookingItemType (**[MS-OXWSCDATA] section 2.2.4.20**)**	For internal use only.<7>

2.2.4.7 t:NonEmptyArrayOfRequestAttachmentIdsType Complex Type

The **NonEmptyArrayOfRequestAttachmentIdsType** complex type represents an array that contains attachment identifiers.

```
<xs:complexType name="NonEmptyArrayOfRequestAttachmentIdsType">
  <xs:choice
    minOccurs="1"
    maxOccurs="unbounded"
  >
    <xs:element name="AttachmentId"
      type="t:RequestAttachmentIdType"
    />
  </xs:choice>
</xs:complexType>
```

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

Element	Type	Description
AttachmentId	**t:RequestAttachmentIdType** (section 2.2.4.9)	Represents an identifier for an attachment.

2.2.4.8 t:ReferenceAttachmentType Complex Type

The **ReferenceAttachmentType** complex type represents a reference attachment. This type extends the **AttachmentType** complex type (section 2.2.4.4). <8>

```
<xs:complexType name="ReferenceAttachmentType">
  <xs:complexContent>
    <xs:extension base="t:AttachmentType">
      <xs:sequence>
        <xs:element name="AttachLongPathName" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="ProviderType" type="xs:string" minOccurs="0" maxOccurs="1"/>
       <xs:element name="ProviderEndpointUrl" type="xs:string" minOccurs="0" maxOccurs="1"/>
       <xs:element name="AttachmentThumbnailUrl" type="xs:string" minOccurs="0"
maxOccurs="1"/>
       <xs:element name="AttachmentPreviewUrl" type="xs:string" minOccurs="0"
maxOccurs="1"/>
       <xs:element name="PermissionType" type="xs:int" minOccurs="0" maxOccurs="1"/>
       <xs:element name="AttachmentIsFolder" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
```


Copyright © 2015 Microsoft Corporation

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

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

Element	Type	Description
AttachLongPathName	**xs:string**([XMLSCHEMA2])	Specifies the URL of the attachment.
ProviderType	**xs:string**	Specifies the provider type.
ProviderEndpointUrl	**xs:string**	Specifies the Url of the provider endpoint.
AttachmentThumbnailUrl	**xs:string**	Specifies the Url of the thumbnail of the attachment.
AttachmentPreviewUrl	**xs:string**	Specifies the Url of the attachment preview.
PermissionType	**xs:int**([XMLSCHEMA2])	Specifies the permission type.
AttachmentIsFolder	**xs:Boolean**([XMLSCHEMA2])	Specifies that the attachment is a folder

2.2.4.9 t:RequestAttachmentIdType Complex Type

The **RequestAttachmentIdType** complex type represents an identifier for an attachment. The **RequestAttachmentIdType** complex type extends the **BaseItemIdType** complex type ([MS-OXWSCDATA] section 2.2.4.15).

```
<xs:complexType name="RequestAttachmentIdType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseItemIdType"
    >
      <xs:attribute name="Id"
        type="xs:string"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table summarizes the set of common XML schema attribute definitions defined by this specification.

Attribute	Type	Description
Id	**xs:string** ([XMLSCHEMA2])	Provides an identifier for an attachment. The maximum length is 512 bytes after base64 decoding.

Copyright © 2015 Microsoft Corporation

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

Copyright © 2015 Microsoft Corporation

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

The Attachment Handling Web Service Protocol defines a single WSDL port type with three operations. The operations enable client implementations to create, delete, and get attachments.

3.1.1 Abstract Data Model

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

The Attachment Handling Web Service Protocol is used to create, delete, and get attachments on items on the account's **mailbox** on the server.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation	Description
CreateAttachment	Creates either an item or file attachment and attaches it to the specified item.
DeleteAttachment	Deletes file and item attachments from an existing item in the server store.
GetAttachment	Retrieves existing attachments on items in the server store.

3.1.4.1 CreateAttachment Operation

The **CreateAttachment** operation creates an item or file attachment on an item in the server store.

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

```
<wsdl:operation name="CreateAttachment">
      <wsdl:input message="tns:CreateAttachmentSoapIn" />
      <wsdl:output message="tns:CreateAttachmentSoapOut" />
</wsdl:operation>
```


Copyright © 2015 Microsoft Corporation

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

```
<wsdl:operation name="CreateAttachment">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateAttachment" />
    <wsdl:input>
        <soap:header message="tns:CreateAttachmentSoapIn" part="Impersonation"
use="literal"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="MailboxCulture"
use="literal"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="RequestVersion"
use="literal"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="TimeZoneContext"
use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="CreateAttachmentResult" use="literal" />
        <soap:header message="tns:CreateAttachmentSoapOut" part="ServerVersion"
use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

The protocol client sends a **CreateAttachmentSoapIn** request WSDL message, and the protocol server responds with a **CreateAttachmentSoapOut** response WSDL message.

An item attachment does not exist as a store item. It only exists as an attachment on an item or another attachment. Item attachments can be retrieved only by using the **GetAttachment** operation request.

3.1.4.1.1 Messages

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

Message	Description
CreateAttachmentSoapIn	Specifies the **SOAP message** that creates an attachment.
CreateAttachmentSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.1.1.1 tns:CreateAttachmentSoapIn Message

The **CreateAttachmentSoapIn** WSDL message specifies the **CreateAttachment** operation request to create an attachment.

```
<wsdl:message name="CreateAttachmentSoapIn">
    <wsdl:part name="request" element="tns:CreateAttachment" />
    <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>
```

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

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

Copyright © 2015 Microsoft Corporation

Part	Element/type	Description
request	**tns:CreateAttachment** (section 3.1.4.1.2.1)	Specifies the **SOAP body** of the request create an attachment.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.4.33)	Specifies a **SOAP header** that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.4.45)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **CreateAttachment** operation request.
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables associating SOAP attributes with the definition.

3.1.4.1.1.2 tns:CreateAttachmentSoapOut Message

The **CreateAttachmentSoapOut** WSDL message specifies the server response to the **CreateAttachment** operation request to create an attachment.

```
<wsdl:message name="CreateAttachmentSoapOut">
    <wsdl:part name="CreateAttachmentResult" element="tns:CreateAttachmentResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part	Element/type	Description
CreateAttachmentResult	**tns:CreateAttachmentResponse** (section 3.1.4.1.2.2)	Specifies the SOAP body of the response to a **CreateAttachment** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

If the request is successful, the **CreateAttachment** operation returns a **CreateAttachmentResponse** element with the **ResponseClass** attribute of the **CreateAttachmentResponseMessage** element set to "Success". The **ResponseCode** element of the **CreateAttachmentResponse** element is set to "NoError".

If the request is unsuccessful, the **CreateAttachment** operation returns a **CreateAttachmentResponse** element with the **ResponseClass** attribute of the **CreateAttachmentResponseMessage** element set to "Error". The **ResponseCode** element of the **CreateAttachmentResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.1.2 Elements

Copyright © 2015 Microsoft Corporation

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

Element	Description
CreateAttachment	Specifies a request to create an attachment.
CreateAttachmentResponse	Specifies the response body content from a request to create an attachment.

3.1.4.1.2.1 CreateAttachment Element

The **CreateAttachment** element specifies the request message for a **CreateAttachment** operation.

```
<xs:element name="CreateAttachment"
  type="m:CreateAttachmentType"
 />
```

3.1.4.1.2.2 CreateAttachmentResponse Element

The **CreateAttachmentResponse** element specifies the response message for a **CreateAttachment** operation (section 3.1.4.1).

```
<xs:element name="CreateAttachmentResponse"
  type="m:CreateAttachmentResponseType"
 />
```

3.1.4.1.3 Complex Types

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

Complex type	Description
CreateAttachmentResponseType	Specifies a response message for the **CreateAttachment** operation.
CreateAttachmentType	Specifies request message for the **CreateAttachment** operation.

3.1.4.1.3.1 m:CreateAttachmentResponseType Complex Type

The **CreateAttachmentResponseType** complex type specifies the response message that is returned by the **CreateAttachment** operation. The **CreateAttachmentResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.18).

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


Copyright © 2015 Microsoft Corporation

3.1.4.1.3.2 m:CreateAttachmentType Complex Type

The **CreateAttachmentType** complex type specifies a request message to attach an item or file to a specified item in the server database. The **CreateAttachmentType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.17).

```
<xs:complexType name="CreateAttachmentType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="ParentItemId"
          type="t:ItemIdType"
         />
        <xs:element name="Attachments"
          type="t:NonEmptyArrayOfAttachmentsType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element	Type	Description
ParentItemId	**t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25)	Identifies the parent item in the server store that contains the attachment. The **ParentItemId** element MUST provide the ID of a store item. The **ParentItemId** element can be retrieved by using the **GetItem** operation ([MS-OXWSCORE] section 3.1.4.4).
Attachments	**t:NonEmptyArrayOfAttachmentsType** ([MS-OXWSCDATA] section 2.2.4.49)	Contains the items or files that are attached to an item in the server store.

3.1.4.2 DeleteAttachment Operation

The **DeleteAttachment** operation deletes attachments from an item in the server store.

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

```
<wsdl:operation name="DeleteAttachment">
    <wsdl:input message="tns:DeleteAttachmentSoapIn" />
    <wsdl:output message="tns:DeleteAttachmentSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the operation.

```
<wsdl:operation name="DeleteAttachment">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteAttachment" />
    <wsdl:input>
        <soap:header message="tns:DeleteAttachmentSoapIn" part="Impersonation"
use="literal"/>
```


Copyright © 2015 Microsoft Corporation

```
        <soap:header message="tns:DeleteAttachmentSoapIn" part="MailboxCulture"
use="literal"/>
        <soap:header message="tns:DeleteAttachmentSoapIn" part="RequestVersion"
use="literal"/>
        <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body parts="DeleteAttachmentResult" use="literal" />
        <soap:header message="tns:DeleteAttachmentSoapOut" part="ServerVersion"
use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

The protocol client sends a **DeleteAttachmentSoapIn** request WSDL message, and the protocol server responds with a **DeleteAttachmentSoapOut** response WSDL message.

An item attachment does not exist as a store item. It only exists as an attachment on an item or another attachment. Item attachments can be retrieved only by using the **GetAttachment** operation request. Before an item attachment can be deleted, it MUST be retrieved from the server.

3.1.4.2.1 Messages

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

Message	Description
DeleteAttachmentSoapIn	Specifies the SOAP message that deletes an attachment.
DeleteAttachmentSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.2.1.1 tns:DeleteAttachmentSoapIn Message

The **DeleteAttachmentSoapIn** WSDL message specifies the **DeleteAttachment** operation request to delete an attachment.

```
<wsdl:message name="DeleteAttachmentSoapIn">
    <wsdl:part name="request" element="tns:DeleteAttachment" />
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

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

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

Part	Element/type	Description
request	**tns:DeleteAttachment** (section 3.1.4.2.2.1)	Specifies the SOAP body of the request to delete an attachment.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.4.33)	Specifies a SOAP header that identifies the user who the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA]	Specifies a SOAP header that identifies the culture

Copyright © 2015 Microsoft Corporation

Part	Element/type	Description
	section 2.2.4.45)	to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **DeleteAttachment** operation request.

3.1.4.2.1.2 tns:DeleteAttachmentSoapOut Message

The **DeleteAttachmentSoapOut** WSDL message specifies the server response to the **DeleteAttachment** operation request to delete an attachment.

```
<wsdl:message name="DeleteAttachmentSoapOut">
    <wsdl:part name="DeleteAttachmentResult" element="tns:DeleteAttachmentResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part	Element/type	Description
DeleteAttachmentResult	**tns:DeleteAttachmentResponse** (section 3.1.4.2.2.2)	Specifies the SOAP body of the response to a **DeleteAttachment** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

If the request is successful, the **DeleteAttachment** operation returns a **DeleteAttachmentResponse** element with the **ResponseClass** attribute of the **DeleteAttachmentResponseMessage** element set to "Success". The **ResponseCode** element of the **DeleteAttachmentResponse** element is set to "NoError".

If the request is unsuccessful, the **DeleteAttachment** operation returns a **DeleteAttachmentResponse** element with the **ResponseClass** attribute of the **DeleteAttachmentResponseMessage** element set to "Error". The **ResponseCode** element of the **DeleteAttachmentResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.2.2 Elements

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

Element	Description
DeleteAttachment	Specifies a request to delete an attachment.
DeleteAttachmentResponse	Specifies the response body content from a request to delete an attachment.

Copyright © 2015 Microsoft Corporation

3.1.4.2.2.1 DeleteAttachment Element

The **DeleteAttachment** element specifies the request message for a **DeleteAttachment** operation.

```
<xs:element name="DeleteAttachment"
  type="m:DeleteAttachmentType"
 />
```

3.1.4.2.2.2 DeleteAttachmentResponse Element

The **DeleteAttachmentResponse** element specifies the response message for a **DeleteAttachment** operation.

```
<xs:element name="DeleteAttachmentResponse"
  type="m:DeleteAttachmentResponseType"
 />
```

3.1.4.2.3 Complex Types

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

Complex type	Description
DeleteAttachmentResponseMessageType	Contains the status and result of a single **DeleteAttachment** operation.
DeleteAttachmentResponseType	Specifies a response message for the **DeleteAttachment** operation.
DeleteAttachmentType	Specifies a request message for the **DeleteAttachment** operation.
RootItemIdType	Identifies the root item of a deleted attachment. Because an attachment can be attached to another attachment, the root item is the store item that owns the entire attachment hierarchy.

3.1.4.2.3.1 m:DeleteAttachmentResponseMessageType Complex Type

The **DeleteAttachmentResponseMessageType** complex type contains the status and result of a single **DeleteAttachment** operation. The **DeleteAttachmentResponseMessageType** complex type extends the **ResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.67).

```
<xs:complexType name="DeleteAttachmentResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="RootItemId"
          type="t:RootItemIdType"
          minOccurs="0"
         />
      </xs:sequence>

    </xs:extension>
```


Copyright © 2015 Microsoft Corporation

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

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

Element	Type	Description
RootItemId	**t:RootItemIdType** (section 3.1.4.2.3.4)	Specifies the parent item of a deleted attachment.

3.1.4.2.3.2 m:DeleteAttachmentResponseType Complex Type

The **DeleteAttachmentResponseType** complex type specifies the response message that is returned by the **DeleteAttachment** operation. The **DeleteAttachmentResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.18).

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

3.1.4.2.3.3 m:DeleteAttachmentType Complex Type

The **DeleteAttachmentType** complex type specifies a request message to delete attachments on an item in the server database. The **DeleteAttachmentType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.17).

```
<xs:complexType name="DeleteAttachmentType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="AttachmentIds"
          type="t:NonEmptyArrayOfRequestAttachmentIdsType"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

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

Element	Type	Description
AttachmentIds	**t:NonEmptyArrayOfRequestAttachmentIdsType** (section 2.2.4.7)	Contains the identifiers of the items or files that are attached to an item in the server store to be deleted.

Copyright © 2015 Microsoft Corporation

3.1.4.2.3.4 t:RootItemIdType Complex Type

The **RootItemIdType** complex type identifies the root item of a deleted attachment and the new change key to the parent item. The **RootItemIdType** complex type extends the **BaseItemIdType** complex type ([MS-OXWSCDATA] section 2.2.4.15).

```
<xs:complexType name="RootItemIdType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseItemIdType"
    >
      <xs:attribute name="RootItemId"
        type="xs:string"
        use="required"
       />
      <xs:attribute name="RootItemChangeKey"
        type="xs:string"
        use="required"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table summarizes the set of common XML schema attribute definitions defined by this specification.

Attribute	Type	Description
RootItemId	**xs:string** ([XMLSCHEMA2])	Identifies the root item of an attachment. The maximum length is 512 bytes after base64 decoding.
RootItemChangeKey	**xs:string**	Identifies the new change key of the root item of an attachment. The maximum length is 512 bytes after base64 decoding.

3.1.4.3 GetAttachment Operation

The **GetAttachment** operation gets attachments from an item in the server store.

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

```
<wsdl:operation name="GetAttachment">
    <wsdl:input message="tns:GetAttachmentSoapIn" />
    <wsdl:output message="tns:GetAttachmentSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification of the operation.

```
<wsdl:operation name="GetAttachment">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetAttachment" />

    <wsdl:input>
        <soap:header message="tns:GetAttachmentSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:GetAttachmentSoapIn" part="MailboxCulture" use="literal"/>
```


Copyright © 2015 Microsoft Corporation

```
            <soap:header message="tns:GetAttachmentSoapIn" part="RequestVersion" use="literal"/>
            <soap:header message="tns:GetAttachmentSoapIn" part="TimeZoneContext" use="literal"/>
            <soap:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="GetAttachmentResult" use="literal" />
            <soap:header message="tns:GetAttachmentSoapOut" part="ServerVersion" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
```

The protocol client sends a **GetAttachmentSoapIn** request WSDL message, and the protocol server responds with a **GetAttachmentSoapOut** response WSDL message.

3.1.4.3.1 Messages

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

Message	Description
GetAttachmentSoapIn	Specifies the SOAP message that gets an attachment.
GetAttachmentSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.3.1.1 tns:GetAttachmentSoapIn Message

The **GetAttachmentSoapIn** WSDL message specifies the **GetAttachment** operation request to get an attachment.

```
<wsdl:message name="GetAttachmentSoapIn">
    <wsdl:part name="request" element="tns:GetAttachment" />
    <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>
```

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

The five parts of the **GetAttachmentSoapIn** message are described in the following table.

Part	Element/type	Description
request	**tns:GetAttachment** (section 3.1.4.3.2.1)	Specifies the SOAP body of the request to get an attachment.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.4.33)	Specifies a SOAP header that identifies the user who the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.4.45)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the **GetAttachment** operation request.

Copyright © 2015 Microsoft Corporation

Part	Element/type	Description
TimeZoneContext	**t:TimeZoneContext** ([MS-OXWSGTZ] section 2.2.3.4)	Specifies a time zone definition and enables associating SOAP attributes with the definition.

3.1.4.3.1.2 tns:GetAttachmentSoapOut Message

The **GetAttachmentSoapOut** WSDL message specifies the server response to the **GetAttachment** operation request to get an attachment.

```
<wsdl:message name="GetAttachmentSoapOut">
    <wsdl:part name="GetAttachmentResult" element="tns:GetAttachmentResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

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

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

Part	Element/type	Description
GetAttachmentResult	**tns:GetAttachmentResponse** (section 3.1.4.3.2.2)	Specifies the SOAP body of the response to a **GetAttachment** operation request.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

If the request is successful, the **GetAttachment** operation returns a **GetAttachmentResponse** element with the **ResponseClass** attribute of the **GetAttachmentResponseMessage** element set to "Success". The **ResponseCode** element of the **GetAttachmentResponse** element is set to "NoError".

If the request is unsuccessful, the **GetAttachment** operation returns a **GetAttachmentResponse** element with the **ResponseClass** attribute of the **GetAttachmentResponseMessage** element set to "Error". The **ResponseCode** element of the **GetAttachmentResponseMessage** element is set to a value of the **ResponseCodeType** simple type, as specified in [MS-OXWSCDATA] section 2.2.5.24.

3.1.4.3.2 Elements

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

Element	Description
GetAttachment	Specifies a request to get an attachment.
GetAttachmentResponse	Specifies the response body content from a request to get an attachment.

3.1.4.3.2.1 GetAttachment Element

The **GetAttachment** element specifies the request message for a **GetAttachment** operation.

```
<xs:element name="GetAttachment"
```


Copyright © 2015 Microsoft Corporation

```
  type="m:GetAttachmentType"
 />
```

3.1.4.3.2.2 GetAttachmentResponse Element

The **GetAttachmentResponse** element specifies the response message to a **GetAttachment** operation.

```
<xs:element name="GetAttachmentResponse"
  type="m:GetAttachmentResponseType"
 />
```

3.1.4.3.3 Complex Types

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

Complex type	Description
GetAttachmentResponseType	Specifies a response message for the **GetAttachment** operation.
GetAttachmentType	Specifies a request message for the **GetAttachment** operation.
AttachmentResponseShapeType	Specifies additional properties for the **GetAttachment** operation to return.

3.1.4.3.3.1 m:GetAttachmentResponseType Complex Type

The **GetAttachmentResponseType** complex type specifies the response message that is returned by the **GetAttachment** operation. The **GetAttachmentResponseType** complex type extends the **BaseResponseMessageType** complex type ([MS-OXWSCDATA] section 2.2.4.18).

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

3.1.4.3.3.2 m:GetAttachmentType Complex Type

The **GetAttachmentType** complex type specifies a request message to get attached items and files on an item in the server database. The **GetAttachmentType** complex type extends the **BaseRequestType** complex type ([MS-OXWSCDATA] section 2.2.4.17).

```
<xs:complexType name="GetAttachmentType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="AttachmentShape"
          type="t:AttachmentResponseShapeType"
```


Copyright © 2015 Microsoft Corporation

```
            minOccurs="0"
           />
          <xs:element name="AttachmentIds"
            type="t:NonEmptyArrayOfRequestAttachmentIdsType"
           />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

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

Element	Type	Description
AttachmentShape	**t:AttachmentResponseShapeType** (section 3.1.4.3.3.3)	Contains additional properties to return for the attachments.
AttachmentIds	**t:NonEmptyArrayOfRequestAttachmentIdsType** (section 2.2.4.7)	Contains the identifiers of the attachments to return in the response.

3.1.4.3.3.3 t:AttachmentResponseShapeType Complex Type

The **AttachmentResponseShapeType** complex type identifies additional properties for the **GetAttachment** operation to return.

```
<xs:complexType name="AttachmentResponseShapeType">
  <xs:sequence>
    <xs:element name="IncludeMimeContent"
      type="xs:boolean"
      minOccurs="0"
     />
    <xs:element name="BodyType"
      type="t:BodyTypeResponseType"
      minOccurs="0"
     />
    <xs:element name="FilterHtmlContent"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
     />
    <xs:element name="AdditionalProperties"
      type="t:NonEmptyArrayOfPathsToElementType"
      minOccurs="0"
     />
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **AttachmentResponseShapeType** complex type.<9>

Element	Type	Description
IncludeMimeContent	**xs:boolean** ([XMLSCHEMA2])	Indicates whether the MIME content of an item or attachment is returned in a response. A text value of "true" indicates that the

Copyright © 2015 Microsoft Corporation

Element	Type	Description
		attachment contains MIME content.<10>
BodyType	**t:BodyTypeResponseType** ([MS-OXWSCDATA] section 2.2.5.1)	Represents the format of the body text in a response.
FilterHtmlContent	**xs:boolean** ([XMLSCHEMA2])	Indicates whether to filter potentially unsafe HTML content from message bodies. A text value of "true" indicates that potentially unsafe HTML content is to be filtered from the attachment.
AdditionalProperties	**t:NonEmptyArrayOfPathsToElementType** ([MS-OXWSCDATA] section 2.2.4.50)	Contains additional properties to return in a response.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client 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.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing Rules

None.

3.2.5 Timer Events

None.

Copyright © 2015 Microsoft Corporation

3.2.6 Other Local Events

None.

Copyright © 2015 Microsoft Corporation

4 Protocol Examples

The following examples show the request and response XML for the Attachment Handling Web Service Protocol operations.

4.1 CreateAttachment Example

The following is an example of a **CreateAttachment** operation, which creates an attachment named Deleteme.txt on the specified message. The item that it is attached to, which is represented by the **ParentItemId** element, is created separately. The **Id** attribute of the **ParentItemId** element has been shortened to preserve readability.

The client constructs the request XML and sends it to the server.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
          xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
          xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:CreateAttachment>
      <m:ParentItemId Id="AAMkAGIwODEy" />
      <m:Attachments>
        <t:FileAttachment>
          <t:Name>Deleteme.txt</t:Name>
          <t:IsInline>false</t:IsInline>
          <t:IsContactPhoto>false</t:IsContactPhoto>
          <t:Content>UGxlYXNlIGRlbGV0ZSB0aGlzIGZpbGUu</t:Content>
        </t:FileAttachment>
      </m:Attachments>
    </m:CreateAttachment>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **Id**, **RootItemId**, and **RootItemChangeKey** attributes of the **AttachmentId** element have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8" ?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
        MinorVersion="1" MajorBuildNumber="63"
        MinorBuildNumber="0"
        Version="Exchange2010"
        xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:CreateAttachmentResponse
        xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:CreateAttachmentResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Attachments>
            <t:FileAttachment>
```


Copyright © 2015 Microsoft Corporation

```
              <t:AttachmentId Id="AAMkAGIwODEyY" RootItemId="AAMkAGIwODEyY"
RootItemChangeKey="CQAAABYAAADL " />
            </t:FileAttachment>
          </m:Attachments>
        </m:CreateAttachmentResponseMessage>
      </m:ResponseMessages>
    </m:CreateAttachmentResponse>
  </s:Body>
</s:Envelope>
```

4.2 DeleteAttachment Example

The following is an example of a **DeleteAttachment** operation. This example deletes an attachment from an item.

The client has to have the identifier for the item that has the attachment. The client constructs the request XML and sends it to the server. The **Id** attribute of the **AttachmentId** element has been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:DeleteAttachment>
      <m:AttachmentIds>
        <t:AttachmentId Id="AAMkAGY4YzQw" />
      </m:AttachmentIds>
    </m:DeleteAttachment>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **RootItemId** and **RootItemChangeKey** attributes of the **RootItemId** element have been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
        MinorVersion="1"
        MajorBuildNumber="63" MinorBuildNumber="0"
        Version="Exchange2010"
xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:DeleteAttachmentResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
          xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:DeleteAttachmentResponseMessage xsi:type="m:DeleteAttachmentResponseMessageType"
            ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:RootItemId RootItemId="AAMkAGY4YzQw=" RootItemChangeKey="CQAAABYAAA" />
        </m:DeleteAttachmentResponseMessage>
```


Copyright © 2015 Microsoft Corporation

```
      </m:ResponseMessages>
    </m:DeleteAttachmentResponse>
  </s:Body>
</s:Envelope>
```

4.3 GetAttachment Example

The following is an example of a **GetAttachment** operation. This example gets an attachment from the server store.

The client constructs the request XML and sends it to the server. The **Id** attribute of the **AttachmentId** element has been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:GetAttachment>
      <m:AttachmentIds>
        <t:AttachmentId Id="AAMkAGY4YzQw" />
      </m:AttachmentIds>
    </m:GetAttachment>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. In this example, there is one attachment in the attachments collection. The **Id** attribute of the **AttachmentId** element has been shortened to preserve readability. Note that the content is base64.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <h:ServerVersionInfo MajorVersion="14"
        MinorVersion="1"
        MajorBuildNumber="63"
        MinorBuildNumber="0"
        Version="Exchange2010"
xmlns:h="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema" />
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <m:GetAttachmentResponse
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:GetAttachmentResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Attachments>
            <t:FileAttachment>
              <t:AttachmentId Id="AAMkAGY4YzQw" />
              <t:Name>Deleteme.txt</t:Name>
              <t:Content>UGxlYXNlIGRlbGV0ZSB0aGlzIHRleHQu</t:Content>
            </t:FileAttachment>
          </m:Attachments>
        </m:GetAttachmentResponseMessage>
```


Copyright © 2015 Microsoft Corporation

```
        </m:ResponseMessages>
      </m:GetAttachmentResponse>
    </s:Body>
  </s:Envelope>
```


Copyright © 2015 Microsoft Corporation

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Copyright © 2015 Microsoft Corporation

6 Appendix A: Full WSDL

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

File name	Description	Section
MS-OXWSATT.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSATT-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	7.1
MS-OXWSATT-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

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

This section contains the contents of the MS-OXWSATT.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:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="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"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
    <xs:include schemaLocation="MS-OXWSATT-messages.xsd" />
    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="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"/>
      <!-- Add global elements and types from types.xsd -->
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="CreateAttachmentSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:CreateAttachment"/>
    <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="CreateAttachmentSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="CreateAttachmentResult" element="tns:CreateAttachmentResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:message name="DeleteAttachmentSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:DeleteAttachment"/>
    <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="DeleteAttachmentSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="DeleteAttachmentResult" element="tns:DeleteAttachmentResponse"/>
```


Copyright © 2015 Microsoft Corporation

```
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:message name="GetAttachmentSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:GetAttachment"/>
    <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="GetAttachmentSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="GetAttachmentResult" element="tns:GetAttachmentResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:portType name="ExchangeServicePortType">
    <wsdl:operation name="CreateAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:input message="tns:CreateAttachmentSoapIn"/>
      <wsdl:output message="tns:CreateAttachmentSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:input message="tns:DeleteAttachmentSoapIn"/>
      <wsdl:output message="tns:DeleteAttachmentSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:input message="tns:GetAttachmentSoapIn"/>
      <wsdl:output message="tns:GetAttachmentSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/"/>
    </wsdl:documentation>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
    <wsdl:operation name="CreateAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateAttachment"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      <wsdl:input>
        <soap:header message="tns:CreateAttachmentSoapIn" part="Impersonation" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="MailboxCulture" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:CreateAttachmentSoapIn" part="TimeZoneContext"
use="literal" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="CreateAttachmentResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:CreateAttachmentSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteAttachment"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      <wsdl:input>
        <soap:header message="tns:DeleteAttachmentSoapIn" part="Impersonation" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:DeleteAttachmentSoapIn" part="MailboxCulture" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:DeleteAttachmentSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
```


Copyright © 2015 Microsoft Corporation

```
        <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="DeleteAttachmentResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:DeleteAttachmentSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAttachment" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetAttachment"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      <wsdl:input>
        <soap:header message="tns:GetAttachmentSoapIn" part="Impersonation" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:GetAttachmentSoapIn" part="MailboxCulture" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:GetAttachmentSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:GetAttachmentSoapIn" part="TimeZoneContext" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="GetAttachmentResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
        <soap:header message="tns:GetAttachmentSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```


Copyright © 2015 Microsoft Corporation

7 Appendix B: Full XML Schema

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

Schema name	Prefix	Section
Messages schema	m:	7.1
Types schema	t:	7.2

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-OXWSATT-types.xsd or MS-OXWSATT-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-OXWSATT-messages.xsd file and information about additional files that this schema file requires to operate correctly.

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

File name	Defining specification
MS-OXWSCDATA-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:tns="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="Exchange2016" id="messages">
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSATT-types.xsd"/>
  <xs:complexType name="AttachmentInfoResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="Attachments" type="t:ArrayOfAttachmentsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="CreateAttachment" type="m:CreateAttachmentType"/>
  <xs:element name="CreateAttachmentResponse" type="m:CreateAttachmentResponseType"/>
  <xs:complexType name="CreateAttachmentResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="CreateAttachmentType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="ParentItemId" type="t:ItemIdType"/>
          <xs:element name="Attachments" type="t:NonEmptyArrayOfAttachmentsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```


Copyright © 2015 Microsoft Corporation

```
  <xs:complexType name="DeleteAttachmentType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="AttachmentIds" type="t:NonEmptyArrayOfRequestAttachmentIdsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="DeleteAttachment" type="m:DeleteAttachmentType"/>
  <xs:complexType name="DeleteAttachmentResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="DeleteAttachmentResponse" type="m:DeleteAttachmentResponseType"/>
  <xs:complexType name="DeleteAttachmentResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="RootItemId" type="t:RootItemIdType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="GetAttachmentType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="AttachmentShape" type="t:AttachmentResponseShapeType"
minOccurs="0"/>
          <xs:element name="AttachmentIds" type="t:NonEmptyArrayOfRequestAttachmentIdsType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetAttachment" type="m:GetAttachmentType"/>
  <xs:complexType name="GetAttachmentResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetAttachmentResponse" type="m:GetAttachmentResponseType"/>
</xs:schema>
```

7.2 Types Schema

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

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

File name	Defining specification
MS-OXWSCONT-types.xsd	[MS-OXWSCONT] section 7
MS-OXWSMSG-types.xsd	[MS-OXWSMSG] section 7
MS-OXWSMTGS-types.xsd	[MS-OXWSMTGS] section 7.2
MS-OXWSPOST-types.xsd	[MS-OXWSPOST] section 7.2

```
<?xml version="1.0" encoding="utf-8"?>
```


Copyright © 2015 Microsoft Corporation

```
<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:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCONT-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSMSG-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSMTGS-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSPOST-types.xsd"/>
  <xs:complexType name="ReferenceAttachmentType">
    <xs:complexContent>
      <xs:extension base="t:AttachmentType">
        <xs:sequence>
          <xs:element name="AttachLongPathName" type="xs:string" minOccurs="0"
maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ArrayOfAttachmentsType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="ItemAttachment" type="t:ItemAttachmentType"/>
      <xs:element name="FileAttachment" type="t:FileAttachmentType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="AttachmentIdType">
    <xs:complexContent>
      <xs:extension base="t:RequestAttachmentIdType">
        <xs:attribute name="RootItemId" type="xs:string" use="optional"/>
        <xs:attribute name="RootItemChangeKey" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="AttachmentResponseShapeType">
    <xs:sequence>
      <xs:element name="IncludeMimeContent" type="xs:boolean" minOccurs="0"/>
      <xs:element name="BodyType" type="t:BodyTypeResponseType" minOccurs="0"/>
      <xs:element name="FilterHtmlContent" type="xs:boolean" minOccurs="0"/>
      <xs:element name="AdditionalProperties" type="t:NonEmptyArrayOfPathsToElementType"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AttachmentType">
    <xs:sequence>
      <xs:element name="AttachmentId" type="t:AttachmentIdType" minOccurs="0"/>
      <xs:element name="Name" type="xs:string" minOccurs="0"/>
      <xs:element name="ContentType" type="xs:string" minOccurs="0"/>
      <xs:element name="ContentId" type="xs:string" minOccurs="0"/>
      <xs:element name="ContentLocation" type="xs:string" minOccurs="0"/>
      <xs:element name="Size" type="xs:int" minOccurs="0"/>
      <xs:element name="LastModifiedTime" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="IsInline" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="FileAttachmentType">
    <xs:complexContent>
      <xs:extension base="t:AttachmentType">
        <xs:sequence>
          <xs:element name="IsContactPhoto" type="xs:boolean" minOccurs="0"/>
          <xs:element name="Content" type="xs:base64Binary" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ItemAttachmentType">
    <xs:complexContent>
      <xs:extension base="t:AttachmentType">
        <xs:choice minOccurs="0">
          <xs:element name="Item" type="t:ItemType"/>
```


Copyright © 2015 Microsoft Corporation

```
            <xs:element name="Message" type="t:MessageType"/>
            <xs:element name="CalendarItem" type="t:CalendarItemType"/>
            <xs:element name="Contact" type="t:ContactItemType"/>
            <xs:element name="MeetingMessage" type="t:MeetingMessageType"/>
            <xs:element name="MeetingRequest" type="t:MeetingRequestMessageType"/>
            <xs:element name="MeetingResponse" type="t:MeetingResponseMessageType"/>
            <xs:element name="MeetingCancellation" type="t:MeetingCancellationMessageType"/>
            <xs:element name="Task" type="t:TaskType"/>
            <xs:element name="PostItem" type="t:PostItemType"/>
            <xs:element name="RoleMember" type="t:RoleMemberItemType"/>
            <xs:element name="Network" type="t:NetworkItemType"/>
            <xs:element name="Person" type="t:AbchPersonItemType"/>
          </xs:choice>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="NonEmptyArrayOfRequestAttachmentIdsType">
      <xs:choice maxOccurs="unbounded">
        <xs:element name="AttachmentId" type="t:RequestAttachmentIdType"/>
      </xs:choice>
    </xs:complexType>
    <xs:complexType name="RequestAttachmentIdType">
      <xs:complexContent>
        <xs:extension base="t:BaseItemIdType">
          <xs:attribute name="Id" type="xs:string" use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="RootItemIdType">
      <xs:complexContent>
        <xs:extension base="t:BaseItemIdType">
          <xs:attribute name="RootItemId" type="xs:string" use="required"/>
          <xs:attribute name="RootItemChangeKey" type="xs:string" use="required"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:schema>
```


Copyright © 2015 Microsoft Corporation

8 Appendix C: Product Behavior

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

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

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

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

<1> Section 2.2.4.4: The **IsInline** element is not supported in Exchange 2007.

<2> Section 2.2.4.5: The **IsContactPhoto** element is not supported in Exchange 2007.

<3> Section 2.2.4.6: In Microsoft Exchange Server 2007 Service Pack 1 (SP1), Exchange 2010, Exchange 2013, and Exchange 2016, generic items will be returned as **MessageType** items.

<4> Section 2.2.4.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **RoleMember** element.

<5> Section 2.2.4.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Network** element.

<6> Section 2.2.4.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Person** element.

<7> Section 2.2.4.6: Exchange 2007, Exchange 2010, and Exchange 2013 do not support the **Booking** element.

<8> Section 2.2.4.8: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ReferenceAttachmentType** complex type. This type was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1).

<9> Section 3.1.4.3.3.3: The **FilterHtmlContent** element is not supported in Exchange 2007.

<10> Section 3.1.4.3.3.3: In Exchange 2007, Exchange 2010, Microsoft Exchange Server 2010 Service Pack 1 (SP1) and Microsoft Exchange Server 2010 Service Pack 2 (SP2), if the **IncludeMimeContent** element is set to **true** in the **AttachmentResponseShapeType** complex type, the MIME content will be returned for attachments of the following class: **IPM.Note**, **IPM.Post**, **IPM.Appointment**. If the **IncludeMimeContent** element is set to **true** and the attachment is not one of the accepted item classes, the **GetAttachmentResponseMessage** element MUST return an **ErrorUnsupportedMimeConversion** response code as specified in [MS-OXWSCDATA] section 2.2.5.24.

Copyright © 2015 Microsoft Corporation

9 Change Tracking

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

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

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

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

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

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

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

Copyright © 2015 Microsoft Corporation

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.4.1 t:ArrayOfAttachmentsType Complex Type	Added details for the ReferenceAttachment child element.	N	Content update.
2.2.4.6 t:ItemAttachmentType Complex Type	Added details for the RoleMember, Network, Person, and Booking child elements.	N	Content update.
2.2.4.6 t:ItemAttachmentType Complex Type	Added new product behavior notes for the RoleMember, Network, Person, and Booking child elements.	N	New product behavior note added.
2.2.4.8 t:ReferenceAttachmentType Complex Type	Added a new section for this complex type.	Y	New content added.
2.2.4.8 t:ReferenceAttachmentType Complex Type	Added a new product behavior note for this complex type.	N	New product behavior note added.

Copyright © 2015 Microsoft Corporation

10 Index

Copyright © 2015 Microsoft Corporation

Copyright © 2015 Microsoft Corporation