

# [MS-OXWSATT]: Attachment Handling Web Service Protocol Specification

---

## 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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). 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](mailto: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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

## Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.0.1	Editorial	Revised and edited the technical content.
02/10/2010	2.0.0	Major	Updated and revised the technical content.
05/05/2010	3.0.0	Major	Updated and revised the technical content.
08/04/2010	4.0	Major	Significantly changed the technical content.
11/03/2010	5.0	Major	Significantly changed the technical content.

# Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Overview	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor Extensible Fields	8
1.9 Standards Assignments	8
<b>2 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Messages	9
2.2.3 Elements	9
2.2.4 Complex Types	9
2.2.4.1 t:ArrayOfAttachmentsType Complex Type	10
2.2.4.2 t:AttachmentIdType Complex Type	10
2.2.4.3 m:AttachmentInfoResponseMessageType Complex Type	11
2.2.4.4 t:AttachmentType Complex Type	11
2.2.4.5 t:FileAttachmentType Complex Type	13
2.2.4.6 t:ItemAttachmentType Complex Type	14
2.2.4.7 t:NonEmptyArrayOfRequestAttachmentIdsType Complex Type	16
2.2.4.8 t:RequestAttachmentIdType Complex Type	16
2.2.4.9 t:RootItemIdType Complex Type	16
2.2.5 Simple Types	17
2.2.6 Attributes	17
2.2.7 Groups	17
2.2.8 Attribute Groups	17
<b>3 Protocol Details</b>	<b>18</b>
3.1 ExchangeServicePortType Server Details	18
3.1.1 Abstract Data Model	18
3.1.2 Timers	18
3.1.3 Initialization	18
3.1.4 Message Processing Events and Sequencing	18
3.1.4.1 CreateAttachment Operation	18
3.1.4.1.1 Messages	19
3.1.4.1.1.1 tns:CreateAttachmentSoapIn Message	19
3.1.4.1.1.2 tns:CreateAttachmentSoapOut Message	20
3.1.4.1.2 Elements	20
3.1.4.1.2.1 CreateAttachment Element	21
3.1.4.1.2.2 CreateAttachmentResponse Element	21
3.1.4.1.3 Complex Types	21
3.1.4.1.3.1 m:CreateAttachmentResponseType Complex Type	21
3.1.4.1.3.2 m:CreateAttachmentType Complex Type	21

3.1.4.2	DeleteAttachment Operation .....	22
3.1.4.2.1	Messages .....	23
3.1.4.2.1.1	tns>DeleteAttachmentSoapIn Message .....	23
3.1.4.2.1.2	tns>DeleteAttachmentSoapOut Message .....	24
3.1.4.2.2	Elements .....	24
3.1.4.2.2.1	DeleteAttachment Element .....	24
3.1.4.2.2.2	DeleteAttachmentResponse Element .....	24
3.1.4.2.3	Complex Types .....	25
3.1.4.2.3.1	m>DeleteAttachmentResponseMessageType Complex Type .....	25
3.1.4.2.3.2	m>DeleteAttachmentResponseType Complex Type .....	25
3.1.4.2.3.3	m>DeleteAttachmentType Complex Type .....	26
3.1.4.3	GetAttachment Operation .....	26
3.1.4.3.1	Messages .....	27
3.1.4.3.1.1	tns:GetAttachmentSoapIn Message .....	27
3.1.4.3.1.2	tns:GetAttachmentSoapOut Message .....	28
3.1.4.3.2	Elements .....	28
3.1.4.3.2.1	GetAttachment Element .....	28
3.1.4.3.2.2	GetAttachmentResponse Element .....	28
3.1.4.3.3	Complex Types .....	28
3.1.4.3.3.1	m:GetAttachmentResponseType Complex Type .....	29
3.1.4.3.3.2	m:GetAttachmentType Complex Type .....	29
3.1.4.3.3.3	t:AttachmentResponseShapeType Complex Type .....	30
3.1.5	Timer Events .....	31
3.1.6	Other Local Events .....	31
<b>4</b>	<b>Protocol Examples .....</b>	<b>32</b>
4.1	CreateAttachment Example .....	32
4.2	DeleteAttachment Example .....	33
4.3	GetAttachment Example .....	34
<b>5</b>	<b>Security .....</b>	<b>36</b>
5.1	Security Considerations for Implementers .....	36
5.2	Index of Security Parameters .....	36
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>37</b>
6.1	WSDL .....	37
6.2	Types Schema .....	40
6.3	Messages Schema .....	42
<b>7</b>	<b>Appendix B: Product Behavior .....</b>	<b>44</b>
<b>8</b>	<b>Change Tracking .....</b>	<b>45</b>
<b>9</b>	<b>Index .....</b>	<b>49</b>

# 1 Introduction

This document specifies the Attachment and Handling Web Service protocol, which is responsible for creating, deleting, and getting attachments on items on the server.

## 1.1 Glossary

The following terms are defined in [\[MS-OXGLOS\]](#):

**attachment**  
**Attachment object**  
**Endpoint (2)**  
**Hypertext Transfer Protocol (HTTP)**  
**Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**  
**message**  
**Message object**  
**Multipurpose Internet Mail Extensions (MIME)**  
**Uniform Resource Identifier (URI)**  
**Uniform Resource Locator (URL)**  
**Web Services Description Language (WSDL)**  
**WSDL message**  
**WSDL port type**  
**XML**  
**XML namespace**  
**XML schema**

The following terms are specific to this document:

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

## 1.2 References

### 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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-OXDSCCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol Specification](#)", April 2008.

[MS-OXWSADISC] Microsoft Corporation, "[Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol Specification](#)", July 2009.

[MS-OXWSCDATA] Microsoft Corporation, "[Common Web Service Data Types](#)", July 2009.

[MS-OXWSCONT] Microsoft Corporation, "[Contacts Web Service Protocol Specification](#)", July 2009.

[MS-OXWSCORE] Microsoft Corporation, "[Core Items Web Service Protocol Specification](#)", July 2009.

- [MS-OXWSGTZ] Microsoft Corporation, "[Get Server Time Zone Web Service Protocol Specification](#)", July 2009.
- [MS-OXWSMSG] Microsoft Corporation, "[E-Mail Message Types Web Service Protocol Specification](#)", July 2009.
- [MS-OXWSMTGS] Microsoft Corporation, "[Calendaring Web Service Protocol Specification](#)", July 2009.
- [MS-OXWSPOST] Microsoft Corporation, "[Post Items Web Service Protocol Specification](#)", July 2009.
- [MS-OXWSTASK] Microsoft Corporation, "[Tasks Web Service Protocol Specification](#)", July 2009.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.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", W3C Note, 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>
- [XMLNS] Bray, T., Hollander, D., Layman, A., Eds., et al., "Namespaces in XML 1.0 (Third Edition)", December 2009, <http://www.w3.org/TR/REC-xml-names/>
- [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., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

## 1.2.2 Informative References

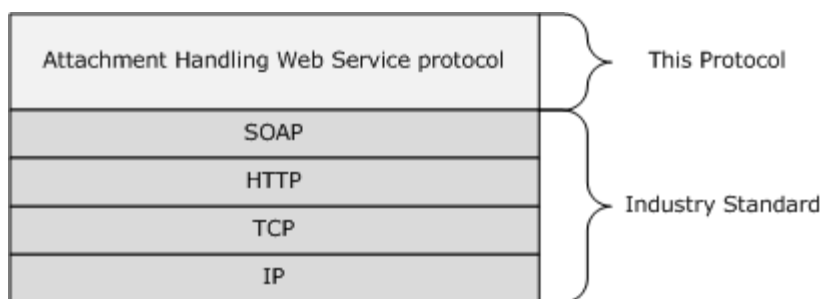
- [MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)", April 2008.

## 1.3 Overview

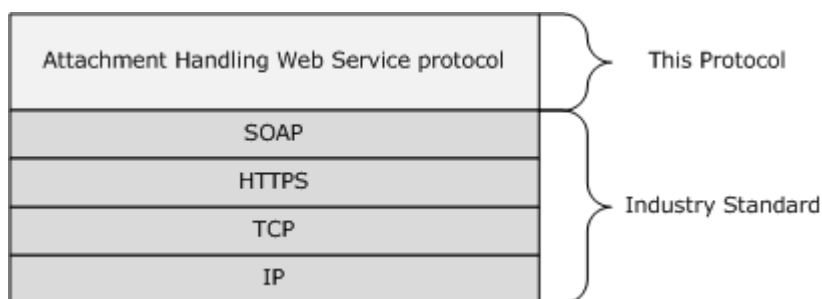
The Attachment Handling Web Service Protocol enables clients to attach objects to **messages**. Clients use **Attachment objects** to associate files, OLE objects, other messages, or binary data with a particular **Message object**. Because Attachment objects are created, maintained, and accessed only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

## 1.4 Relationship to Other Protocols

The Attachment Handling Web Service protocol uses SOAP over **HTTP** and SOAP over **HTTPS**, as shown in the following figures.



**Figure 1: Attachment Handling Web Service protocol HTTP stack**



**Figure 2: Attachment Handling Web Service protocol HTTPS stack**

## 1.5 Prerequisites/Preconditions

The **endpoint (2) URL** that is returned by either the Autodiscover Publishing and Lookup SOAP-Based Web Service protocol [\[MS-OXWSADISC\]](#) or the Autodiscover Publishing and Lookup protocol [\[MS-OXDSCLI\]](#) is a required argument for forming the HTTP request to the Web server that hosts this protocol. The operations that this protocol uses cannot be accessed unless the correct endpoint (2) that services the target **mailbox** is identified for the HTTP Web requests that target the Voice Mail Settings Web Service protocol operations.

## 1.6 Applicability Statement

The protocol specified in this document is applicable to environments that attach items to messages.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1.
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Capability Negotiation:** This protocol does not perform explicit capability negotiation. This protocol relies on the Autodiscover Publishing and Lookup protocol [\[MS-OXDSCLI\]](#) to identify protocol versions and service endpoints that are applicable to a client.

## **1.8 Vendor Extensible Fields**

None.

## **1.9 Standards Assignments**

None.



## 2 Messages

### 2.1 Transport

The SOAP version supported is SOAP 1.1. For details, see [\[SOAP1.1\]](#).

### 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	<a href="http://schemas.xmlsoap.org/wsdl/soap/">http://schemas.xmlsoap.org/wsdl/soap/</a>	<a href="#">[SOAP1.1]</a>
tns	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	
s	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a>
targetNamespace	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>
t	<a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a>	

#### 2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

#### 2.2.3 Elements

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

#### 2.2.4 Complex Types

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

Complex type	Description
<b>ArrayOfAttachmentsType</b>	Represents an array of types based on <b>attachments</b> to the item.
<b>AttachmentIdType</b>	Represents the item identifier and new change key of an item after an attachment has been attached.

Complex type	Description
<b>AttachmentInfoResponseMessageType</b>	Contains status and response data for attachments.
<b>AttachmentType</b>	Represents an attachment.
<b>FileAttachmentType</b>	Represents a file that is attached to an item in the server store.
<b>ItemAttachmentType</b>	Represents an item that is attached to another item.
<b>NonEmptyArrayOfRequestAttachmentIdsType</b>	Represents an array of attachment <b>identifiers</b> .
<b>RequestAttachmentIdType</b>	Represents identifier for an attachment.
<b>RootItemIdType</b>	Identifies the root item of a deleted attachment.

### 2.2.4.1 t:ArrayOfAttachmentsType Complex Type

The **ArrayOfAttachmentsType** complex type represents an array of types based upon attachments to the item. The **ArrayOfAttachmentsType** complex type is only used 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:choice>
</xs:complexType>
```

Child Elements

Element	Type	Description
ItemAttachment	<a href="#">t:ItemAttachmentType</a>	Specifies an item that is attached to another item.
FileAttachment	<a href="#">t:FileAttachmentType</a>	Specifies a file 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.

```
<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>

```

#### Attributes

Name	Type	Description
RootItemId	<a href="#">xs:string</a>	Represents the unique identifier of the root store item to which the attachment is attached. The <b>xs:string</b> type is defined in <a href="#">[XMLSCHEMA2]</a> .
RootItemChangeKey	<a href="#">xs:string</a>	Represents the change key of the root store item to which the attachment is attached.

#### 2.2.4.3 m:AttachmentInfoResponseMessageType Complex Type

The **AttachmentInfoResponseMessageType** complex type contains status and response data for attachments.

```

<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>

```

#### Child Elements

Element	Type	Description
Attachments	<a href="#">t:ArrayOfAttachmentsType</a>	Represents an array of types based on attachments on the item. The ArrayOfAttachmentsType complex type is only used 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"
      type="xs:boolean"
      minOccurs="0"
      maxOccurs="1"
    />
  </xs:sequence>
</xs:complexType>

```

## Child Elements

Element	Type	Description
AttachmentId	<a href="#">t:AttachmentIdType</a>	The attachment identifier.
Name	<a href="#">xs:string</a>	Specifies the descriptive name of the attachment. The <b>xs:string</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
ContentType	xs:string	Specifies the <b>MIME</b> type of the attachment content. For example, "text/enriched", "text/html", "text/plain",

Element	Type	Description
		"text/rfc822-headers", "text/richtext", or "text/sgml".
ContentId	<a href="#">xs:string</a>	Specifies the unique object identifier for an attachment.
ContentLocation	<a href="#">xs:string</a>	The <b>URI</b> 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	<a href="#">xs:int</a>	Provides an informational estimate of the size, in bytes, of the complete body of the item. The <b>xs:int</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
LastModifiedTime	<a href="#">xs:dateTime</a>	The day and time that this item was last changed. The <b>xs:dateTime</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
IsInline	<a href="#">xs:boolean</a>	A Boolean value that indicates whether the attachment is an inline attachment. The <IsInline> element is set to <b>true</b> to indicate that the attachment appears inline within an item. The <b>xs:boolean</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .

The **ItemAttachmentType** complex type (section [2.2.4.6](#)) and the **FileAttachmentType** complex type (section [2.2.4.5](#)) extend the **AttachmentType** complex type.

### 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.

```
<xs:complexType name="FileAttachmentType">
  <xs:complexContent>
    <xs:extension name="FileAttachmentType"
      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>
```

#### Child Elements

Element	Type	Description
IsContactPhoto	<a href="#">xs:boolean</a>	A value that indicates whether this attachment is a contact photo.

Element	Type	Description
		The <b>xs:boolean</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
Content	<a href="#">xs:base64Binary</a>	Represents the base64-encoded contents of the file attachment. The <b>xs:base64Binary</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .

#### 2.2.4.6 t:ItemAttachmentType Complex Type

The **ItemAttachmentType** complex type represents an item that is attached to another item. The **ItemAttachmentType** complex type extends the **AttachmentType** complex type.

```

<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"
        />
        <xs:element name="MeetingCancellation"
          type="t:MeetingCancellationMessageType"
        />
        <xs:element name="Task"
          type="t:TaskType"
        />
        <xs:element name="PostItem"
          type="t:PostItemType"
        />
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## Child Elements

Element	Type	Description
Item	t:ItemType	Represents a generic item in the server store. The <b>ItemType</b> complex type is specified in <a href="#">[MS-OXWScore]</a> section 2.2.3.6.
Message	t:MessageType	Represents a server e-mail message. The <b>MessageType</b> complex type is specified in <a href="#">[MS-OXWSMSG]</a> section 2.2.4.1.
CalendarItem	t:CalendarItemType	Represents a calendar item. The <b>CalendarItemType</b> complex type is specified in <a href="#">[MS-OXWSMTGS]</a> section 2.2.3.4.
Contact	t:ContactItemType	Represents a contact item. The <b>ContactItemType</b> complex type is specified in <a href="#">[MS-OXWSCont]</a> section 2.2.4.1.
MeetingMessage	t:MeetingMessageType	Represents a meeting message in the server store. The <b>MeetingMessageType</b> complex type is specified in <a href="#">[MS-OXWSMTGS]</a> section 2.2.3.11.
MeetingRequest	t:MeetingRequestMessageType	Represents a meeting request in the server store. The <b>MeetingRequestMessageType</b> complex type is specified in <a href="#">[MS-OXWSMTGS]</a> section 2.2.3.12.
MeetingResponse	t:MeetingResponseMessageType	Represents a meeting response in the server store. The <b>MeetingResponseMessageType</b> complex type is specified in <a href="#">[MS-OXWSMTGS]</a> section 2.2.3.13.
MeetingCancellation	t:MeetingCancellationMessageType	Represents a meeting cancellation in the server store. The <b>MeetingCancellationMessageType</b> is specified in <a href="#">[MS-OXWSMTGS]</a> section 2.2.3.10.
Task	t:TaskType	Represents a task in the server store. The <b>TaskType</b> complex type is specified in <a href="#">[MS-OXWSTASK]</a> section 2.2.3.3.
PostItem	<a href="#">t:PostItemType</a>	Represents a post item in the server store. The <b>PostItemType</b> complex type is specified in <a href="#">[MS-OXWSPOST]</a> section 2.2.3.1.

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

#### Child Elements

Element	Type	Description
AttachmentId	<a href="#">t:RequestAttachmentIdType</a>	Represents an identifier for an attachment.

### 2.2.4.8 t:RequestAttachmentIdType Complex Type

The **RequestAttachmentIdType** complex type represents an identifier for an attachment.

```
<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>
```

#### Attributes

Name	Type	Description
Id	<a href="#">xs:string</a>	Provides an identifier for an attachment. The <b>xs:string</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .

The **AttachmentIdType** complex type (section [2.2.4.2](#)) extends the **RequestAttachmentIdType** complex type.

### 2.2.4.9 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.



```

<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>

```

## Attributes

Name	Type	Description
RootItemId	<a href="#">xs:string</a>	Identifies the root item of an attachment. The <b>xs:string</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
RootItemChangeKey	<a href="#">xs:string</a>	Identifies the new change key of the root item of an attachment.

### 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.

## 3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

### 3.1 ExchangeServicePortType Server Details

This protocol defines a single WSDL port type that contains a set of operations, as listed in the following table.

Operation	Description
<a href="#">CreateAttachment</a>	Creates either an item or file attachment and attaches it to the specified item.
<a href="#">DeleteAttachment</a>	Deletes file and item attachments from an existing item in the server store.
<a href="#">GetAttachment</a>	Retrieves existing attachments on items in the server store.

#### 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.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing

This protocol includes the operations that are listed in the following table.

Operation	Description
<b>CreateAttachment</b>	Creates either an item or file attachment and attaches it to the specified item.
<b>DeleteAttachment</b>	Deletes file and item attachments from an existing item in the server store.
<b>GetAttachment</b>	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.

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

</wsdl:operation>

#### Request

Message Format	Description
<a href="#">tns:CreateAttachmentSoapIn</a>	Specifies the SOAP message that creates an attachment on the item.

#### Response

Message Format	Description
<a href="#">tns:CreateAttachmentSoapOut</a>	Specifies the SOAP message that is returned by the server in response.

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

A successful **CreateAttachment** operation request 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".

### 3.1.4.1.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
<b>CreateAttachmentSoapIn</b>	Specifies the request that creates an attachment.
<b>CreateAttachmentSoapOut</b>	Specifies the response to a request that creates an attachment.

#### 3.1.4.1.1.1 tns:CreateAttachmentSoapIn Message

The **CreateAttachmentSoapIn** WSDL message specifies the **CreateAttachment** operation request.

```
<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 five parts of the **CreateAttachmentSoapIn** message are described in the following table.

Part name	Element/type	Description
request	<tns:CreateAttachment>	Specifies the request message.

Part name	Element/type	Description
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC3066]</a> . The <MailboxCulture> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the schema version for the <b>CreateAttachment</b> operation request. The <RequestServerVersion> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.
TimeZoneContext	<t:TimeZoneContext>	Specifies a time zone definition and enables associating SOAP attributes with the definition. The <TimeZoneContext> element is specified in <a href="#">[MS-OXWSGTZ]</a> section 2.2.4.4.

### 3.1.4.1.1.2 tns:CreateAttachmentSoapOut Message

The **CreateAttachmentSoapOut** WSDL message specifies the **CreateAttachment** operation response.

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

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

Part name	Element/type	Description
CreateAttachmentResult	<tns:CreateAttachmentResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.8.

### 3.1.4.1.2 Elements

The following table lists the XML schema elements that are specific to this operation.

Element	Description
<CreateAttachment>	Specifies a request to create an attachment.
<CreateAttachmentResponse>	Specifies the response to a request to create an attachment.

### 3.1.4.1.2.1 CreateAttachment Element

The <CreateAttachment> element defines a request to create an attachment to an item in the server store.

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

### 3.1.4.1.2.2 CreateAttachmentResponse Element

The <CreateAttachmentResponse> element specifies the response message for a **GetAttachment** operation.

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

### 3.1.4.1.3 Complex Types

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

Complex type	Description
<b>CreateAttachmentResponseType</b>	Specifies a response to a <b>CreateAttachment</b> operation request.
<b>CreateAttachmentType</b>	Specifies a <b>CreateAttachment</b> operation request.

#### 3.1.4.1.3.1 m:CreateAttachmentResponseType Complex Type

The **CreateAttachmentResponseType** complex type specifies the complex type that is returned by the **CreateAttachment** operation.

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

#### 3.1.4.1.3.2 m:CreateAttachmentType Complex Type

The **CreateAttachmentType** complex type represents a request to attach an item or file to a specified item in the server database.

```
<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>

```

#### Child Elements

Element	Type	Description
ParentItemId	t:ItemIdType	Identifies the parent item in the server store that contains the attachment. The <ParentItemId> element MUST provide the ID of a real store item. The <ParentItemId> element can be retrieved by using the <b>GetItem</b> operation. The <b>ItemIdType</b> complex type is specified in <a href="#">[MS-OXWSCORE]</a> section 2.2.3.13.
Attachments	t:NonEmptyArrayOfAttachmentsType	Element contains the items or files that are attached to an item in the server store. The <b>NonEmptyArrayOfAttachmentsType</b> complex type is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.39.

#### 3.1.4.2 DeleteAttachment Operation

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

```

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

```

#### Request

Message Format	Description
<a href="#">tns:DeleteAttachmentSoapIn</a>	Specifies the SOAP message that deletes an attachment from the item.

#### Response

Message Format	Description
<a href="#">tns:DeleteAttachmentSoapOut</a>	Specifies the SOAP message that is returned by the server in response.

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

A successful **DeleteAttachment** operation request 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".

### 3.1.4.2.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
<b>DeleteAttachmentSoapIn</b>	Specifies the request that deletes an attachment.
<b>DeleteAttachmentSoapOut</b>	Specifies the response to a request that deletes an attachment.

#### 3.1.4.2.1.1 tns:DeleteAttachmentSoapIn Message

The **DeleteAttachmentSoapIn** WSDL message specifies the **DeleteAttachment** operation request.

```
<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 four parts of the **DeleteAttachmentSoapIn** message are described in the following table.

Part name	Element/type	Description
request	<tns:DeleteAttachment>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC3066]</a> . The <MailboxCulture> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the target schema version for the <b>DeleteAttachment</b> operation request.

Part name	Element/type	Description
		The <RequestServerVersion> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.

### 3.1.4.2.1.2 tns:DeleteAttachmentSoapOut Message

The **DeleteAttachmentSoapOut** WSDL message specifies the **DeleteAttachment** operation response.

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

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

Part name	Element/type	Description
DeleteAttachmentResult	<tns:DeleteAttachmentResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.8.

### 3.1.4.2.2 Elements

The following table lists the XML schema elements that are specific to this operation.

Element	Description
<DeleteAttachment>	Specifies a request to delete an attachment.
<DeleteAttachmentResponse>	Specifies the response to a request to delete an attachment.

#### 3.1.4.2.2.1 DeleteAttachment Element

The <DeleteAttachment> element is the root element in a request to delete an attachment from the server store.

```
<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 lists the XML schema complex type definitions that are specific to this operation.

Complex type	Description
<b>DeleteAttachmentResponseMessageType</b>	The parent item of a deleted attachment.
<b>DeleteAttachmentResponseType</b>	Specifies a response to a <b>DeleteAttachment</b> operation request.
<b>DeleteAttachmentType</b>	Specifies a <b>DeleteAttachment</b> operation request.

#### 3.1.4.2.3.1 m:DeleteAttachmentResponseMessageType Complex Type

The **DeleteAttachmentResponseMessageType** complex type contains the status and result of a single **DeleteAttachment** operation.

```

<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>

```

Child Elements

Element	Type	Description
RootItemId	<a href="#">t:RootItemIdType</a>	The parent item of a deleted attachment.

#### 3.1.4.2.3.2 m:DeleteAttachmentResponseType Complex Type

The **DeleteAttachmentResponseType** complex type specifies the complex type that is returned by the **DeleteAttachment** operation.

```

<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 represents a request to delete an attachment on an item in the server database.

```

<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>

```

#### Child Elements

Element	Type	Description
AttachmentIds	<a href="#">t:NonEmptyArrayOfRequestAttachmentIdsType</a>	Contains the items or files that are attached to an item in the server store to be deleted.

### 3.1.4.3 GetAttachment Operation

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

```

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

```

#### Request

Message Format	Description
<a href="#">tns:GetAttachmentSoapIn</a>	Specifies the SOAP message that gets an attachment from the item.

#### Response

Message Format	Description
<a href="#">tns:GetAttachmentSoapOut</a>	Specifies the SOAP message that is returned by the server in response.

A successful **GetAttachment** operation request 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".

### 3.1.4.3.1 Messages

The following WSDL message definitions are specific to this operation.

Message	Description
<b>GetAttachmentSoapIn</b>	Specifies the request that gets an attachment.
<b>GetAttachmentSoapOut</b>	Specifies the response to a request that gets an attachment.

#### 3.1.4.3.1.1 tns:GetAttachmentSoapIn Message

The **GetAttachmentSoapIn** WSDL message specifies the **GetAttachment** operation request.

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

Part name	Element/type	Description
request	<tns:GetAttachment>	Specifies the request message.
Impersonation	<t:ExchangeImpersonation>	Specifies the identifier of the account to impersonate. The <ExchangeImpersonation> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.
MailboxCulture	<t:MailboxCulture>	Specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC3066]</a> . The <MailboxCulture> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.5.
RequestVersion	<t:RequestServerVersion>	Specifies the target schema version for the <b>GetAttachment</b> operation request. The <RequestServerVersion> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.2.
TimeZoneContext	<t:TimeZoneContext>	Specifies a time zone definition and enables associating SOAP attributes with the definition. The <TimeZoneContext> element is specified in <a href="#">[MS-OXWSGTZ]</a> section 2.2.4.4.

### 3.1.4.3.1.2 tns:GetAttachmentSoapOut Message

The **GetAttachmentSoapOut** WSDL message specifies the **GetAttachment** operation response.

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

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

Part name	Element/type	Description
GetAttachmentResult	<tns:GetAttachmentResponse>	Specifies the response message.
ServerVersion	<t:ServerVersionInfo>	Specifies the server version for the response. The <ServerVersionInfo> element is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.8.

### 3.1.4.3.2 Elements

The following table lists the XML schema elements that are specific to this operation.

Element	Description
<GetAttachment>	Specifies a request to get an attachment.
<GetAttachmentResponse>	Specifies the response to a request to get an attachment.

#### 3.1.4.3.2.1 GetAttachment Element

The <GetAttachment> element defines a request to create an attachment to an item in the server store.

```
<xs:element name="GetAttachment"
  type="m:GetAttachmentType"
/>
```

#### 3.1.4.3.2.2 GetAttachmentResponse Element

The <GetAttachmentResponse> element defines a response to a **GetAttachment** operation request.

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

### 3.1.4.3.3 Complex Types

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

Complex type	Description
<b>GetAttachmentResponseType</b>	Specifies a response to a <b>GetAttachment</b> operation request.
<b>GetAttachmentType</b>	Specifies a <b>GetAttachment</b> operation request.
<b>AttachmentResponseShapeType</b>	Specifies additional properties for the <b>GetAttachment</b> operation to return.

### 3.1.4.3.3.1 m:GetAttachmentResponseType Complex Type

The <GetAttachmentResponseType> complex type contains the response message from the **GetAttachment** operation request.

```
<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 represents a request to get attached items and files on an item in the server database.

```
<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:complexType>
```

#### Child Elements

Element	Type	Description
AttachmentShape	<a href="#">t:AttachmentResponseShapeType</a>	Contains additional properties to return for the attachments.

Element	Type	Description
AttachmentIds	<a href="#">t:NonEmptyArrayOfRequestAttachmentIdsType</a>	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"
    />
    <xs:element name="AdditionalProperties"
      type="t:NonEmptyArrayOfPathsToElementType"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
```

#### Child Elements

Element	Type	Description
IncludeMimeContent	<a href="#">xs:boolean</a>	Indicates whether the MIME content of an item or attachment is returned in a response. The <b>xs:boolean</b> type is specified in <a href="#">[XMLSCHEMA2]</a> .
BodyType	t:BodyTypeResponseType	Represents the format of the body text in a response. The <b>BodyTypeResponseType</b> simple type is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.2.1.
FilterHtmlContent	<a href="#">xs:boolean</a>	Indicates whether to filter potentially unsafe HTML content from message bodies.
AdditionalProperties	t:NonEmptyArrayOfPathsToElementType	Contains Additional properties to return in a response. The

Element	Type	Description
		<b>NonEmptyArrayOfPathsToElementType</b> e complex type is specified in <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.40.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.

## 4 Protocol Examples

The following examples show the request and response XML that is used to perform the specific operations.

### 4.1 CreateAttachment Example

The following is an example of a **CreateAttachment**. This example 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.

```
<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.

```
<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>
            <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>
    </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>
</EwsLogEntry>

```

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-encoded of the attachment file.

```

<?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>UGx1YXNlIGRlbGV0ZSB0aGlzIHRleHQw</t:Content>
          </t:FileAttachment>
        </m:Attachments>
      </m:GetAttachmentResponseMessage>
    </m:ResponseMessages>
  </m:GetAttachmentResponse>
</s:Body>
</s:Envelope>
</EwsLogEntry>
```

## **5 Security**

### **5.1 Security Considerations for Implementers**

The Attachment Handling Web Service protocol does not use any additional security mechanisms.

### **5.2 Index of Security Parameters**

None.

## 6 Appendix A: Full WSDL

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

File name	Description	Section
MS-OXWSATT.wsdl	Contains the WSDL for the implementation of this protocol	<a href="#">6.1</a>
MS-OXWSATT-types.xsd	Contains the XML schema type definitions that are used in this protocol.	<a href="#">6.2</a>
MS-OXWSATT-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	<a href="#">6.3</a>

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 with these files.

### 6.1 WSDL

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:s="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2010"
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="Exchange2010"
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: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:portType>

```

```

        </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: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"/>
        <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: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/">
        </wsi:claim>
        </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/" />
        <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>

```

## 6.2 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	<a href="#">[MS-OXWSCONT]</a> section 6.2
MS-OXWSMSG-types.xsd	<a href="#">[MS-OXWSMSG]</a> section 6.2
MS-OXWSMTGS-types.xsd	<a href="#">[MS-OXWSMTGS]</a> section 6.2
MS-OXWSPOST-types.xsd	<a href="#">[MS-OXWSPOST]</a> section 6.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="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"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010" 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="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:sequence>
  </xs:complexType>
</xs:schema>
```



```

        <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"/>
                <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: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:complexType>
</xs:schema>

```

### 6.3 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	<a href="#">[MS-OXWSCDATA]</a> section 6.2

```

<?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="Exchange2010" id="messages">
  <!--      CHANGE THE SCHEMA LOCATION TO REFLECT THE TYPES XSD ASSOCIATED WITH THIS
DOCUMENT
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
  schemaLocation="types.xsd"/>
  <xs:include/>
-->
  <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>

```

```

        </xs:complexContent>
    </xs:complexType>
    <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 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Exchange Server 2007
- Microsoft® Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. 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 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 product does not follow the prescription.

## 8 Change Tracking

This section identifies changes that were made to the [MS-OXWSATT] protocol document between the August 2010 and November 2010 releases. 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- Changes made for template compliance.
- 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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

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

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

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

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
<a href="#">1 Introduction</a>	53990 Updated the introductory text.	N	Content updated.
<a href="#">1.1 Glossary</a>	53990 Added the term "Attachment object" to the list of terms that are defined in [MS-OXGLOS].	N	Content updated.
<a href="#">1.1 Glossary</a>	Added "endpoint" to the list of terms that are defined in [MS-OXGLOS].	N	Content updated.
<a href="#">1.2.1 Normative References</a>	Added [MS-OXDCLI], [MS-OXWSADISC], [MS-OXWSCDATA], [MS-OXWSCORE], [MS-OXWSTASK] and [MS-OXWSGTZ] to the list of normative references.	N	New content added.
<a href="#">1.3 Overview</a>	53990 Updated the overview information.	N	Content updated.
<a href="#">1.7 Versioning and Capability Negotiation</a>	Added information about capability negotiation.	N	Content updated.
<a href="#">2.2.2 Messages</a>	Updated the section title.	N	Editorially updated.
<a href="#">2.2.4.4 t:AttachmentType Complex Type</a>	Added cross-references.	N	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
<a href="#">2.2.4.5</a> <a href="#">t:FileAttachmentType Complex Type</a>	Added cross-references.	N	Content updated.
<a href="#">2.2.4.8</a> <a href="#">t:RequestAttachmentIdType Complex Type</a>	Added a cross-reference for xs:string.	N	Content updated.
<a href="#">2.2.4.9</a> <a href="#">t:RootItemIdType Complex Type</a>	Added a cross-reference for xs:string.	N	Content updated.
<a href="#">3.1.1</a> <a href="#">Abstract Data Model</a>	Updated the Abstract Data Model content.	N	Content updated.
<a href="#">3.1.4.1</a> <a href="#">CreateAttachment Operation</a>	Added information about a successful request.	N	New content added.
<a href="#">3.1.4.1.1</a> <a href="#">Messages</a>	Added a table that lists and describes the WSDL messages.	N	Content updated.
<a href="#">3.1.4.1.1.1</a> <a href="#">tns:CreateAttachmentSoapIn Message</a>	Updated the introductory text and added the WSDL syntax.	N	Content updated.
<a href="#">3.1.4.1.1.2</a> <a href="#">tns:CreateAttachmentSoapOut Message</a>	Updated the introductory text and added the WSDL syntax.	N	Content updated.
<a href="#">3.1.4.1.2</a> <a href="#">Elements</a>	Added a table that lists the elements.	N	Content updated.
<a href="#">3.1.4.1.3</a> <a href="#">Complex Types</a>	Added a table that lists the complex types.	N	Content updated.
<a href="#">3.1.4.2</a> <a href="#">DeleteAttachment Operation</a>	Added information about a successful request.	Y	New content added.
<a href="#">3.1.4.2.1</a> <a href="#">Messages</a>	Added a table that lists and describes the WSDL messages.	N	Content updated.
<a href="#">3.1.4.2.1.1</a> <a href="#">tns&gt;DeleteAttachmentSoapIn Message</a>	Update the introductory text and added the WSDL syntax.	N	New content added.
<a href="#">3.1.4.2.1.2</a> <a href="#">tns&gt;DeleteAttachmentSoapOut Message</a>	Updated the introductory text and added the WSDL syntax.	N	Content updated.
<a href="#">3.1.4.2.2</a> <a href="#">Elements</a>	Added a table that lists the elements.	Y	New content added.
<a href="#">3.1.4.2.3</a>	Updated the descriptions of the complex	N	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
<a href="#">Complex Types</a>	types.		updated.
<a href="#">3.1.4.3 GetAttachment Operation</a>	Added information about a successful request.	Y	New content added.
<a href="#">3.1.4.3.1 Messages</a>	Added a table that lists and describes the WSDL messages.	N	Content updated.
<a href="#">3.1.4.3.1.1 tns:GetAttachmentSoapIn Message</a>	Updated the introductory text and added WSDL syntax.	N	New content added.
<a href="#">3.1.4.3.1.2 tns:GetAttachmentSoapOut Message</a>	53989 Updated the introduction and added the message syntax.	N	Content updated.
<a href="#">3.1.4.3.2 Elements</a>	Added a table that lists the elements.	Y	New content added.
<a href="#">3.1.4.3.3 Complex Types</a>	Added a table that lists the complex types.	N	Content updated.
<a href="#">3.1.4.3.3.1 m:GetAttachmentResponseType Complex Type</a>	Updated the description of the GetAttachmentResponseType complex type.	N	Content updated.
<a href="#">3.1.4.3.3.3 t:AttachmentResponseShapeType Complex Type</a>	55414 Updated the child elements table to include cross-references for the types.	N	Content updated.
<a href="#">Z Appendix B: Product Behavior</a>	57835 Added Exchange 2007 to the list of applicable products.	Y	New product behavior note added.



## 9 Index

### A

[Applicability](#) 7

### C

[Capability negotiation](#) 7

[Change tracking](#) 45

Common message syntax

[complex types](#) 9

### F

[Full WSDL](#) 37

### G

[Glossary](#) 5

### I

[Introduction](#) 5

### M

Messages

[overview](#) 9

[syntax](#) 9

[transport](#) 9

### O

[Overview \(synopsis\)](#) 6

### P

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 44

### R

References

[informative](#) 6

[normative](#) 5

[Relationship to other protocols](#) 6

### S

Security

[implementer considerations](#) 36

[overview](#) 36

[parameter index](#) 36

Server

[abstract data model](#) 18

[initialization](#) 18

[local events](#) 31

[message processing](#) 18

[sequencing rules](#) 18

[timer events](#) 31

[timers](#) 18

Sever

[overview](#) 18

[Standards assignments](#) 8

### T

[Tracking changes](#) 45

### V

[Vendor-extensible fields](#) 8

[Versioning](#) 7