

# [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.mspx>). 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 [iplq@microsoft.com](mailto:iplq@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.

# Table of Contents

<b>1 Introduction</b> .....	<b>6</b>
1.1 Glossary.....	6
1.2 References.....	6
1.2.1 Normative References .....	6
1.2.2 Informative References .....	7
1.3 Protocol Overview .....	7
1.4 Relationship to Other Protocols.....	7
1.5 Prerequisites/Preconditions.....	7
1.6 Applicability Statement.....	8
1.7 Versioning and Capability Negotiation.....	8
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 Simple Types .....	9
2.2.3 Complex Types.....	9
2.2.3.1 t:ArrayOfAttachmentsType Complex Type.....	10
2.2.3.2 t:AttachmentIdType Complex Type.....	10
2.2.3.3 m:AttachmentInfoResponseMessageType Complex Type.....	11
2.2.3.4 t:AttachmentType Complex Type .....	11
2.2.3.5 t:FileAttachmentType Complex Type.....	13
2.2.3.6 t:ItemAttachmentType Complex Type.....	13
2.2.3.7 t:NonEmptyArrayOfRequestAttachmentIdsType Complex Type.....	15
2.2.3.8 t:RequestAttachmentIdType Complex Type .....	15
2.2.3.9 t:RootItemIdType Complex Type .....	16
2.2.4 Elements.....	16
2.2.5 Attributes.....	16
2.2.6 Groups.....	16
2.2.7 Attribute Groups .....	16
2.2.8 Message Syntax.....	16
<b>3 Protocol Details</b> .....	<b>17</b>
3.1 ExchangeServicePortType Server Details .....	17
3.1.1 Abstract Data Model.....	17
3.1.2 Timers .....	17
3.1.3 Initialization .....	17
3.1.4 Message Processing Events and Sequencing .....	17
3.1.4.1 CreateAttachment.....	18
3.1.4.1.1 Simple Types .....	18
3.1.4.1.2 Complex Type .....	18
3.1.4.1.2.1 m:CreateAttachmentResponseType Complex Type.....	18
3.1.4.1.2.2 m:CreateAttachmentType Complex Type .....	18
3.1.4.1.3 Elements.....	19
3.1.4.1.3.1 CreateAttachment Element.....	19
3.1.4.1.3.2 CreateAttachmentResponse Element .....	19
3.1.4.1.4 Attributes.....	20
3.1.4.1.5 Groups.....	20

3.1.4.1.6	Attribute Groups .....	20
3.1.4.1.7	Messages .....	20
3.1.4.1.7.1	tns:CreateAttachmentSoapIn .....	20
3.1.4.1.7.2	tns:CreateAttachmentSoapOut .....	20
3.1.4.2	DeleteAttachment .....	20
3.1.4.2.1	Simple Types .....	21
3.1.4.2.2	Complex Types.....	21
3.1.4.2.2.1	m>DeleteAttachmentResponseMessageType Complex Type .....	21
3.1.4.2.2.2	m>DeleteAttachmentResponseType Complex Type .....	22
3.1.4.2.2.3	m>DeleteAttachmentType Complex Type .....	22
3.1.4.2.3	Elements.....	23
3.1.4.2.3.1	DeleteAttachment Element.....	23
3.1.4.2.3.2	DeleteAttachmentResponse Element.....	23
3.1.4.2.4	Attributes.....	23
3.1.4.2.5	Groups.....	23
3.1.4.2.6	Attribute Groups .....	23
3.1.4.2.7	Messages .....	23
3.1.4.2.7.1	tns>DeleteAttachmentSoapIn.....	23
3.1.4.2.7.2	tns>DeleteAttachmentSoapOut .....	24
3.1.4.3	GetAttachment.....	24
3.1.4.3.1	Simple Types .....	24
3.1.4.3.2	Complex Type .....	24
3.1.4.3.2.1	m:GetAttachmentResponseType Complex Type .....	25
3.1.4.3.2.2	m:GetAttachmentType Complex Type.....	25
3.1.4.3.2.3	t:AttachmentResponseShapeType Complex Type.....	25
3.1.4.3.3	Elements.....	26
3.1.4.3.3.1	GetAttachment Element .....	26
3.1.4.3.3.2	GetAttachmentResponse Element.....	27
3.1.4.3.4	Attributes.....	27
3.1.4.3.5	Groups.....	27
3.1.4.3.6	Attribute Groups .....	27
3.1.4.3.7	Messages .....	27
3.1.4.3.7.1	tns:GetAttachmentSoapIn .....	27
3.1.4.3.7.2	tns:GetAttachmentSoapOut.....	27
3.1.5	Timer Events.....	28
3.1.6	Other Local Events .....	28
3.2	Client Details.....	28
3.2.1	Abstract Data Model.....	28
3.2.2	Timers .....	28
3.2.3	Initialization .....	28
3.2.4	Message Processing Events and Sequencing .....	28
3.2.5	Timer Events.....	28
3.2.6	Other Local Events .....	28
<b>4</b>	<b>Protocol Examples .....</b>	<b>29</b>
4.1	CreateAttachment Example .....	29
4.2	DeleteAttachment Example .....	30
4.3	GetAttachment Example.....	31
<b>5</b>	<b>Security.....</b>	<b>33</b>
5.1	Security Considerations for Implementers.....	33
5.2	Index of Security Parameters .....	33

<b>6 Appendix A: Full WSDL</b> .....	<b>34</b>
6.1 WSDL.....	34
6.2 Types Schema.....	37
6.3 Messages Schema.....	39
<b>7 Appendix B: Product Behavior</b> .....	<b>41</b>
<b>8 Change Tracking</b> .....	<b>42</b>
<b>9 Index</b> .....	<b>45</b>

# 1 Introduction

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

An Attachment object is used by a client 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 sub-objects.

## 1.1 Glossary

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

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

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

[MS-OXWSMSG] Microsoft Corporation, "[E-Mail Message Types Web Service Protocol Specification](#)", July 2009.

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

[MS-OXWSPOST] Microsoft Corporation, "[Post Items 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", RFC 3066, January 2001, <http://www.ietf.org/rfc/rfc3066.txt>.

[SOAP1.1] Box, D., 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>.

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/REC-xml-names/>.

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

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

### 1.2.2 Informative References

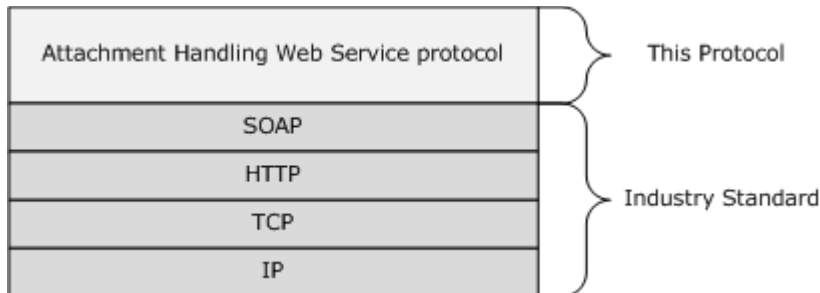
None.

### 1.3 Protocol Overview

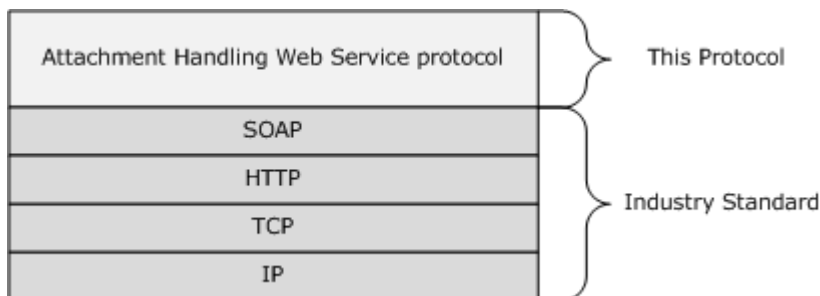
The Attachment Handling Web Service Protocol provides clients with the basis for attaching objects to messages

### 1.4 Relationship to Other Protocols

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



**Figure 1: SOAP over HTTP**



**Figure 2: SOAP over HTTPS**

### 1.5 Prerequisites/Preconditions

None.

## 1.6 Applicability Statement

The protocol specified in this document is applicable to environments that attach items to messages, using Exchange Web Services.

## 1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

**Supported Transports:** This protocol uses SOAP 1.1, as specified in [2.1](#).

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

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

**Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).

**Capability Negotiation:** None.

## 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** 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
<b>soap</b>	<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>	[MS-OXWSATT]
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>	[MS-OXWSATT]
<b>wsdl</b>	<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>	[MS-OXWSATT]

#### 2.2.2 Simple Types

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

#### 2.2.3 Complex Types

The following set of common XML schema complex type definitions that are defined by this specification.

Complex Type	Description
<a href="#">ArrayOfAttachmentsType</a>	Represents an array of types based upon attachments to the item.
<a href="#">AttachmentIdType</a>	Represents the item identifier and new change key of an item after an attachment has been attached.
<a href="#">AttachmentInfoResponseMessageType</a>	Contains status and response data for attachments.
<a href="#">AttachmentType</a>	Represents an attachment.
<a href="#">FileAttachmentType</a>	Represents a file that is attached to an item in the server

Complex Type	Description
	store.
<a href="#">ItemAttachmentType</a>	Represents an item that is attached to another item.
<a href="#">NonEmptyArrayOfRequestAttachmentIdsType</a>	Represents an array of attachment identifiers.
<a href="#">RequestAttachmentIdType</a>	Represents identifier for an attachment.
<a href="#">RootItemIdType</a>	Identifies the root item of a deleted attachment.

### 2.2.3.1 t:ArrayOfAttachmentsType Complex Type

The **ArrayOfAttachmentsType** complex type represents an array of types based upon attachments to the item. The ArrayOfAttachmentsType 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>	An item that is attached to another item.
FileAttachment	<a href="#">t:FileAttachmentType</a>	A file that is attached to another item.

### 2.2.3.2 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"
      />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

        use="optional"
    />
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

#### Attributes

Name	Type	Description
RootItemId	xs:string	Represents the unique identifier of the root store item to which the attachment is attached.
RootItemChangeKey	xs:string	Represents the change key of the root store item to which the attachment is attached.

### 2.2.3.3 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 upon attachments to the item. The ArrayOfAttachmentsTypes is only used in the response message.

### 2.2.3.4 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"
    />
  </xs:sequence>
</xs:complexType>

```

```

    />
    <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>

```

## Child Elements

Element	Type	Description
AttachmentId	<a href="#">t:AttachmentIdType</a>	The attachment identifier.
Name	xs:string	Specifies the descriptive name of the attachment.
ContentType	xs:string	Specifies the Multipurpose Internet Mail Extensions (MIME) type of the attachment content.
ContentId	xs:string	Specifies the unique object identifier for an attachment.
ContentLocation	xs:string	The Uniform Resource Identifier (URI) that corresponds to the location of the content of the attachment.
Size	xs:int	Provides an informational estimate of the size, in bytes, of the complete body of the item.
LastModifiedTime	xs:dateTime	The day and time that this item was last changed.
IsInline	xs:boolean	A Boolean value that indicates whether the attachment is an inline attachment.

The [ItemAttachmentType](#) and [FileAttachmentType](#) extend the **AttachmentType**.

The `ContentType` element is used to indicate the Mime content type of the attachment. For example "text/enriched", "text/html", text/plain, "text/rfc822-headers", "text/richtext", or "text/sgml".

The `ContentLocation` element can be used to associate an attachment with a URL that defines its location on the Web.

The `IsInline` element is set to `True` to indicate that the attachment appears in line within an item.

### 2.2.3.5 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	xs:boolean	A value indicating whether this attachment is a contact photo.
Content	xs:base64Binary	Represents the base64-encoded contents of the file attachment.

### 2.2.3.6 t:ItemAttachmentType Complex Type

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

```
<xs:complexType name="ItemAttachmentType">
  <xs:complexContent>
    <xs:extension
      base="t:AttachmentType"
    >
      <xs:choice
        minOccurs="0"
      >
        <xs:element name="Item"
          type="t:ItemType"
        />
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

<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	<a href="#">t:ItemType</a>	Represents a generic item in the server store. An Item is the base class for <a href="#">TaskType</a> , <a href="#">CalendarItemType</a> , <a href="#">ContactItemType</a> , and <a href="#">MessageType</a> complex types.
Message	t:MessageType	Represents a server e-mail message.
CalendarItem	t:CalendarItemType	Represents a calendar item.
Contact	t:ContactItemType	Represents a contact item.
MeetingMessage	<a href="#">t:MeetingMessageType</a>	Represents a meeting message in the server store.
MeetingRequest	<a href="#">t:MeetingRequestMessageType</a>	Represents a meeting request in the server store.
MeetingResponse	<a href="#">t:MeetingResponseMessageType</a>	Represents a meeting response in the server store.
MeetingCancellation	<a href="#">t:MeetingCancellationMessageType</a>	Represents a meeting cancellation in the

Element	Type	Description
		server store.
Task	t:TaskType	Represents a task in the server store.
PostItem	<a href="#">t:PostItemType</a>	Represents a post item in the server store.

### 2.2.3.7 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.

### 2.2.3.8 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	xs:string	Provides an identifier for an attachment.

The [AttachmentIdType](#) extends the **RequestAttachmentIdType**.

### 2.2.3.9 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	xs:string	Identifies the root item of an attachment.
RootItemChangeKey	xs:string	Identifies the new change key of the root item of an attachment.

### 2.2.4 Elements

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

### 2.2.5 Attributes

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

### 2.2.6 Groups

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

### 2.2.7 Attribute Groups

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

### 2.2.8 Message Syntax

This specification does not define any common XML schema message 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

The Attachment Handling Web Service Protocol has three operations to manipulate attachments.

Operation	Description
<a href="#">CreateAttachment</a>	The operation creates either an item or file attachment and attaches it to the specified item.
<a href="#">DeleteAttachment</a>	The operation is used to delete file and item attachments from an existing item in the server store.
<a href="#">GetAttachment</a>	The operation is used to retrieve 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 listed in the following table.

Operation	Description
<a href="#">CreateAttachment</a>	The operation creates either an item or file attachment and attaches it to the specified item.
<a href="#">DeleteAttachment</a>	The operation is used to delete file and item attachments from an existing item in the server store.
<a href="#">GetAttachment</a>	The operation is used to retrieve existing attachments on items in the server store.

### 3.1.4.1 CreateAttachment

This protocol uses the **CreateAttachment** operation to create an attachment to an item in the server store. This operation creates either an item or file attachment and attaches it to the specified item.

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

Request

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

Response

Message Format	Description
<a href="#">tns:CreateAttachmentSoapOut</a>	Specifies the SOAP message 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](#) request.

#### 3.1.4.1.1 Simple Types

This specification does not define any specific XML schema simple type definitions for this operation.

#### 3.1.4.1.2 Complex Type

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

##### 3.1.4.1.2.1 m:CreateAttachmentResponseType Complex Type

The **CreateAttachmentResponseType** complex type specifies the complex type returned by the [CreateAttachment](#) operation.

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

##### 3.1.4.1.2.2 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	<a href="#">t:ItemIdType</a>	Identifies the parent item in the server store that contains the attachment.
Attachments	<a href="#">t:NonEmptyArrayOfAttachmentsType</a>	Element contains the items or files that are attached to an item in the server store.

The ParentItemId element MUST provide the Id of a real store item. The ParentItemId can be retrieved by using the [GetItem](#) operation.

### 3.1.4.1.3 Elements

The following XML schema element definitions are specific to this operation.

#### 3.1.4.1.3.1 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"
/>

```

#### 3.1.4.1.3.2 CreateAttachmentResponse Element

The <CreateAttachmentResponse> element specifies the response message for a [GetAttachment](#) operation.

```

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

```

#### 3.1.4.1.4 Attributes

This specification does not define any specific XML schema attribute definitions for this operation.

#### 3.1.4.1.5 Groups

This specification does not define any specific XML schema group definitions for this operation.

#### 3.1.4.1.6 Attribute Groups

This specification does not define any specific XML schema attribute group definitions for this operation.

#### 3.1.4.1.7 Messages

The following **WSDL message** definitions are specific to this operation.

##### 3.1.4.1.7.1 tns:CreateAttachmentSoapIn

The [CreateAttachmentSoapIn](#) message contains five parts.

Part Name	Element/Type	Description
request	<a href="#">tns:CreateAttachment</a>	This part specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	The identifier of the account to impersonate.
MailboxCulture	<a href="#">t:MailboxCulture</a>	This part specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	This part specifies the schema version for the <a href="#">CreateAttachment</a> request.
TimeZoneContext	<a href="#">t:TimeZoneContext</a>	Specifies a time zone definition and enables associating SOAP attributes with the definition.

##### 3.1.4.1.7.2 tns:CreateAttachmentSoapOut

The [CreateAttachmentSoapOut](#) message contains two parts.

Part Name	Element/Type	Description
CreateAttachmentResult	<a href="#">tns:CreateAttachmentResponse</a>	This part specifies the response.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	This part specifies the server version for the response.

#### 3.1.4.2 DeleteAttachment

This protocol uses the **DeleteAttachment** operation to delete 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 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](#) request. To delete an attachment, you MUST first retrieve the item from the server.

#### 3.1.4.2.1 Simple Types

This specification does not define any specific XML schema simple type definitions for this operation.

#### 3.1.4.2.2 Complex Types

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

ComplexType	Description
<a href="#">m:DeleteAttachmentResponseMessageType</a>	The parent item of a deleted attachment.
<a href="#">m:DeleteAttachmentResponseType</a>	Specifies the complex type returned by the <a href="#">DeleteAttachment</a> operation.
<a href="#">m:DeleteAttachmentType</a>	Represents a request to delete an attachment on an item in the server database.

##### 3.1.4.2.2.1 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.

#### 3.1.4.2.2.2 m:DeleteAttachmentResponseType Complex Type

The **DeleteAttachmentResponseType** complex type specifies the complex type returned by the [DeleteAttachment](#) operation.

```

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

```

#### 3.1.4.2.2.3 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>	Element contains the items or files that are attached to an item in the

Element	Type	Description
		server store to be deleted.

### 3.1.4.2.3 Elements

The following XML schema element definitions are specific to this operation.

#### 3.1.4.2.3.1 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"
/>
```

#### 3.1.4.2.3.2 DeleteAttachmentResponse Element

The <DeleteAttachmentResponse> element specifies the response message for a [DeleteAttachment](#) operation.

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

#### 3.1.4.2.4 Attributes

This specification does not define any specific XML schema attribute definitions for this operation.

#### 3.1.4.2.5 Groups

This specification does not define any specific XML schema group definitions for this operation.

#### 3.1.4.2.6 Attribute Groups

This specification does not define any specific XML schema group definitions for this operation.

#### 3.1.4.2.7 Messages

The following WSDL message definitions are specific to this operation.

##### 3.1.4.2.7.1 tns:DeleteAttachmentSoapIn

The [DeleteAttachmentSoapIn](#) message contains four parts.

Part Name	Element/Type	Description
request	<a href="#">tns:DeleteAttachment</a>	This part specifies the request.

Part Name	Element/Type	Description
Impersonation	<a href="#">t:ExchangeImpersonation</a>	The identifier of the account to impersonate.
MailboxCulture	<a href="#">t:MailboxCulture</a>	This part specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC 3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	This part specifies the schema version for the <a href="#">DeleteAttachment</a> request

### 3.1.4.2.7.2 tns:DeleteAttachmentSoapOut

The [DeleteAttachmentSoapOut](#) message contains two parts.

Part Name	Element/Type	Description
DeleteAttachmentResult	<a href="#">tns:DeleteAttachmentResponse</a>	This part specifies the response.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	This part specifies the server version for the response.

### 3.1.4.3 GetAttachment

This protocol uses the **GetAttachment** operation to get 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 returned by the server in response.

#### 3.1.4.3.1 Simple Types

This specification does not define any specific XML schema simple type definitions for this operation.

#### 3.1.4.3.2 Complex Type

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



### 3.1.4.3.2.1 m:GetAttachmentResponseType Complex Type

The <GetAttachmentResponseType> complex type represents a request to get attached items and files on an item in the server database.

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

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

#### Child Elements

Element	Type	Description
AttachmentShape	<a href="#">t:AttachmentResponseShapeType</a>	The element contains additional properties to return for the attachments.
AttachmentIds	<a href="#">t:NonEmptyArrayOfRequestAttachmentIdsType</a>	Element contains the identifiers of the attachments to return in the response.

### 3.1.4.3.2.3 t:AttachmentResponseShapeType Complex Type

The **AttachmentResponseShapeType** complex type identifies additional properties for the [GetAttachment](#) element 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	xs:boolean	A Boolean value that indicates whether the MIME content of an item or attachment is returned in a response.
BodyType	<a href="#">t:BodyTypeResponseType</a>	The format of body text in a response.
FilterHtmlContent	xs:boolean	Indicates whether or not to filter potentially unsafe HTML content from message bodies.
AdditionalProperties	<a href="#">t:NonEmptyArrayOfPathsToElementType</a>	Additional properties to return in a response.

### 3.1.4.3.3 Elements

The following XML schema element definitions are specific to this operation.

#### 3.1.4.3.3.1 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"
/>

```

### 3.1.4.3.3.2 GetAttachmentResponse Element

The <GetAttachmentResponse> element defines a response to a [GetAttachment](#) operation.

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

### 3.1.4.3.4 Attributes

This specification does not define any specific XML schema attribute definitions for this operation.

### 3.1.4.3.5 Groups

This specification does not define any specific XML schema group definitions for this operation.

### 3.1.4.3.6 Attribute Groups

This specification does not define any specific XML schema attribute group definitions for this operation.

### 3.1.4.3.7 Messages

The following WSDL message definitions are specific to this operation.

#### 3.1.4.3.7.1 tns:GetAttachmentSoapIn

The [GetAttachmentSoapIn](#) message contains five parts.

Part Name	Element/Type	Description
request	<a href="#">tns:GetAttachment</a>	This part specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	The identifier of the account to impersonate.
MailboxCulture	<a href="#">t:MailboxCulture</a>	This part specifies the culture to use for accessing the server. The cultures are defined by <a href="#">[RFC3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	This part specifies the schema version for the <a href="#">GetAttachment</a> request
TimeZoneContext	<a href="#">t:TimeZoneContext</a>	Specifies a time zone definition and enables associating SOAP attributes with the definition.

#### 3.1.4.3.7.2 tns:GetAttachmentSoapOut

The [GetAttachmentSoapOut](#) message contains two parts.

Part Name	Element/Type	Description
GetAttachmentResult	<a href="#">tns:GetAttachmentResponse</a>	This part specifies the response.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	This part specifies the server version for the

Part Name	Element/Type	Description
		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 that are 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

None.

### 3.2.5 Timer Events

None.

### 3.2.6 Other Local Events

None.

## 4 Protocol Examples

The following examples, [CreateAttachment Example](#), [DeleteAttachment Example](#) and [GetAttachment Example](#) show the request and response XML to perform the specific operations.

### 4.1 CreateAttachment Example

The following is example of a [CreateAttachment](#) Operation with the [CreateAttachmentType](#) complex type. This example creates an attachment named Deleteme.txt to the specified message. The item that it is attached to, the ParentItemId, is created separately. The ParentItemId Id attribute 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>UGxlyXNlIGRlbGV0ZSB0aGlzIGZpbGÜu</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 AttachmentId Id, RootItemId, and RootItemChangeKey attributes 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 with the [DeleteAttachmentType](#) complex type. This example deletes an attachment from an item.

The client MUST have the identifier for the item that has the attachment. The client constructs the request XML and sends it to the server. The AttachmentId Id attribute 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 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 with the [GetAttachmentType](#) complex type. This example gets an attachment from the server store.

The client constructs the request XML and sends it to the server. The AttachmentId attribute 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 AttachmentId Id attribute 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>UGx1YXNlIGRlbGV0ZSB0aGlzIHRleHQ</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 following table lists the **XML** files that are required to implement the functionality that is specified in this document. The contents of each file are included in this section.

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

These files **MUST** be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSATT-types.xsd or MS-OXWSATT-messages.xsd schemas **MUST** be placed in the common folder with the 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"/>
        </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 file listed in the following table. For the schema file to operate correctly, this file MUST be present in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

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

```
<?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: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-types.xsd includes the file listed in the following table. For the schema file to operate correctly, this file MUST be in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

Defining protocol	File name
<a href="#">[MS-OXWSCDATA]: Common Exchange Server Web Service Data Types Specification</a>	MS-OXWSCDATA - messages.xsd

```

<?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="AttachmentInfoResponseType">
    <xs:complexContent>
      <xs:extension base="m:ResponseType">
        <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:BaseResponseType"/>
    </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:extension>
    </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:BaseResponseType" />
    </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:BaseResponseType" />
    </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 product versions. References to product versions include released service packs.

- 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.

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 made to [MS-OXWSATT] protocol documentation between November 2009 and February 2010 releases. Changes are classed as major, minor, or editorial.

**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.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

**Minor** changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

**Editorial** changes apply to grammatical, formatting, and style issues.

**No changes** means that the document is identical to its last release.

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

- New content added.
- Content update.
- 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 always have the revision 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.

Changes are listed in the following table. If you need further information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
<a href="#">1.1 Glossary</a>	53793 Removed the terms "SOAP body", "SOAP fault", and "SOAP header" from the list of terms that are defined in [MS-OXGLOS].	N	Content removed.
<a href="#">2.2.1 Namespaces</a>	53792 Updated link for [XMLNS].	N	Content update.
<a href="#">2.2.1 Namespaces</a>	53793 Updated term tagging for XML namespace.	N	Content update.
<a href="#">2.2.3.4 t:AttachmentType Complex Type</a>	53990 Added remarks.Added ContentType description.	N	New content added.
<a href="#">2.2.3.8 t:RequestAttachmentIdType Complex Type</a>	53990 Added remarks.	N	New content added.
<a href="#">3.1.4.1 CreateAttachment</a>	53990 Added remarks.Modified description. Inserted code snippet.	N	New content added.
<a href="#">3.1.4.1.2.2 m:CreateAttachmentType Complex Type</a>	53990 Added remarks and additional descriptions to the section.	N	Editorially updated.
<a href="#">3.1.4.2 DeleteAttachment</a>	53990 Modified Mark tag. Inserted code snippet.	N	New content added.
<a href="#">3.1.4.3 GetAttachment</a>	53990 Modified Mark tag. Added code snippet.	N	New content added.
<a href="#">3.1.4.3.2.2 m:GetAttachmentType</a>	53990 Updated complexType name.	N	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
<a href="#">Complex Type</a>			
4 <a href="#">Protocol Examples</a>	53990 Added three attachments to the document.	N	New content added.
4.1 <a href="#">CreateAttachment Example</a>	53990 Added new topic.	Y	New content added.
4.2 <a href="#">DeleteAttachment Example</a>	53990 Added new topic.	Y	New content added.
4.3 <a href="#">GetAttachment Example</a>	53990 Added new topic.	Y	New content added.

## 9 Index

### A

[Applicability](#) 8

### C

[Capability negotiation](#) 8

[Change tracking](#) 42

#### Client

[abstract data model](#) 28

[initialization](#) 28

[local events](#) 28

[message processing](#) 28

[overview](#) 28

[sequencing rules](#) 28

[timer events](#) 28

[timers](#) 28

### F

[Full WSDL](#) 34

### G

[Glossary](#) 6

### I

[Introduction](#) 6

### M

#### Messages

[overview](#) 9

[syntax](#) 9

[transport](#) 9

### O

[Overview](#) 7

### P

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 41

### R

#### References

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

### S

#### Security

[implementer considerations](#) 33

[overview](#) 33

[parameter index](#) 33

#### Server

[abstract data model](#) 17

[initialization](#) 17

[local events](#) 28

[message processing](#) 17

[sequencing rules](#) 17

[timer events](#) 28

[timers](#) 17

#### Sever

[overview](#) 17

[Standards assignments](#) 8

### T

[Tracking changes](#) 42

### V

[Vendor-extensible fields](#) 8

[Versioning](#) 8