

[MS-OXWSMSG]: E-Mail Message Types 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 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	2.0.0	Major	Updated and revised the technical content.
02/10/2010	3.0.0	Major	Updated and revised the technical content.
05/05/2010	3.1.0	Minor	Updated 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

1 Introduction	5
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	6
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Common Message Syntax	8
2.2.1 Namespaces	8
2.2.2 Message Syntax	8
2.2.3 Elements	8
2.2.4 Complex Types	8
2.2.4.1 t:MessageType Complex Type	8
2.2.5 Simple Types	11
2.2.5.1 t:MessageDispositionType Simple Type	12
2.2.6 Attributes	12
2.2.7 Groups	12
2.2.8 Attribute Groups	12
3 Protocol Details	13
3.1 ExchangeServicePortType Server Details	13
3.1.1 Abstract Data Model	13
3.1.2 Timers	13
3.1.3 Initialization	13
3.1.4 Message Processing Events and Sequencing	13
3.1.4.1 CreateItem	14
3.1.4.2 GetItem	14
3.1.4.3 UpdateItem	15
3.1.4.4 DeleteItem	16
3.1.4.5 MoveItem	16
3.1.4.6 CopyItem	17
3.1.4.7 SendItem	17
3.1.5 Timer Events	18
3.1.6 Other Local Events	18
4 Protocol Examples	19
4.1 Create Message Example	19
4.2 Get Message Example	20
4.3 Update Message Example	21
4.4 Delete Message Example	22
4.5 Move Message Example	23
4.6 Copy Message Example	25

4.7 Send Message Example	26
5 Security.....	28
5.1 Security Considerations for Implementers.....	28
5.2 Index of Security Parameters	28
6 Appendix A: Full WSDL.....	29
6.1 WSDL.....	29
6.2 Types Schema.....	32
7 Appendix B: Product Behavior	34
8 Change Tracking.....	35
9 Index	37

1 Introduction

This document specifies the E-Mail Message Types Web Service protocol, which represents an e-mail message in a mailbox.

1.1 Glossary

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

attachment
Junk E-mail folder
message
Multipurpose Internet Mail Extensions (MIME)
SOAP message
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. 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-OXWSCDATA] Microsoft Corporation, "[Common Web Service Data Types](#)", July 2009.

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

[MS-OXWSFOLD] Microsoft Corporation, "[Folders and Folder Permissions 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>

[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 E-Mail Message Types Web Service protocol provides clients with the ability to create, update, and delete e-mail items on the server. Clients create e-mail items using the CreateItem operation, or get the properties of an existing task item using the GetItem operation. E-mails can also be sent, updated, deleted, or copied on the server using SendItem, UpdateItem, DeleteItem, and CopyItem respectively.

1.4 Relationship to Other Protocols

The E-Mail Message Types Web Service protocol uses SOAP over **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as shown in the following figures.

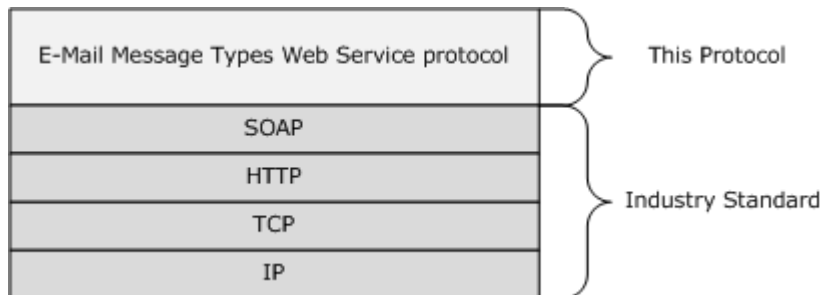


Figure 1: SOAP over HTTP

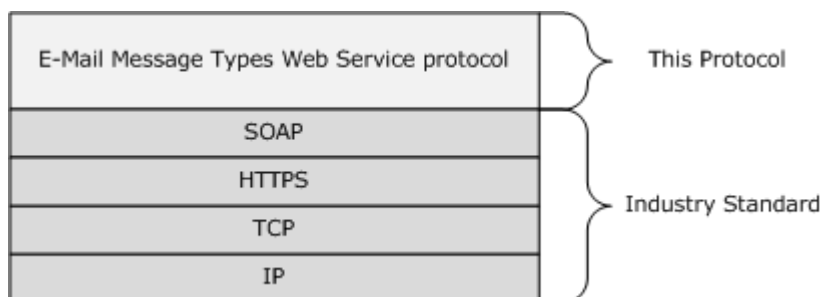


Figure 2: SOAP over HTTPS

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The protocol specified in this document is applicable to environments that copy, create, delete, get, send, or update e-mails 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 section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol 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 specified in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and **Web Services Description Language (WSDL)**, as specified 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 namespaces** prefix is implementation-specific and is not significant for interoperability.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSMSG]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSMSG]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	[MS-OXWSMSG]

2.2.2 Message Syntax

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

2.2.3 Elements

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

2.2.4 Complex Types

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

ComplexType	Description
t:MessageType	Represents a server e-mail message in a mailbox.

2.2.4.1 t:MessageType Complex Type

The **MessageType** complex type represents a server e-mail message in a mailbox.


```

<xs:complexType name="MessageType">
  <xs:complexContent>
    <xs:extension
      base="t:ItemType"
    >
      <xs:sequence>
        <xs:element name="Sender"
          type="t:SingleRecipientType"
          minOccurs="0"
        />
        <xs:element name="ToRecipients"
          type="t:ArrayOfRecipientsType"
          minOccurs="0"
        />
        <xs:element name="CcRecipients"
          type="t:ArrayOfRecipientsType"
          minOccurs="0"
        />
        <xs:element name="BccRecipients"
          type="t:ArrayOfRecipientsType"
          minOccurs="0"
        />
        <xs:element name="IsReadReceiptRequested"
          type="xs:boolean"
          minOccurs="0"
        />
        <xs:element name="IsDeliveryReceiptRequested"
          type="xs:boolean"
          minOccurs="0"
        />
        <xs:element name="ConversationIndex"
          type="xs:base64Binary"
          minOccurs="0"
        />
        <xs:element name="ConversationTopic"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="From"
          type="t:SingleRecipientType"
          minOccurs="0"
        />
        <xs:element name="InternetMessageId"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="IsRead"
          type="xs:boolean"
          minOccurs="0"
        />
        <xs:element name="IsResponseRequested"
          type="xs:boolean"
          minOccurs="0"
        />
        <xs:element name="References"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="ReplyTo"

```

```

        type="t:ArrayOfRecipientsType"
        minOccurs="0"
    />
    <xs:element name="ReceivedBy"
        type="t:SingleRecipientType"
        minOccurs="0"
    />
    <xs:element name="ReceivedRepresenting"
        type="t:SingleRecipientType"
        minOccurs="0"
    />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Child Elements

Element	Type	Description
Sender	t:SingleRecipientType	The <Sender> element gets or sets the sender of a message. This element is optional. This is a read/write element.
ToRecipients	t:ArrayOfRecipientsType	The <ToRecipients> element gets or sets a collection of recipients of an e-mail. This element is required for sending a message. This is a read/write element.
CcRecipients	t:ArrayOfRecipientsType	The <CcRecipients> element gets or sets a collection of recipients that receive a carbon copy (Cc) of an e-mail. This element is optional. This is a read/write element.
BccRecipients	t:ArrayOfRecipientsType	The <BccRecipients> element gets or sets a collection of recipients that receive a blind carbon copy (Bcc) of an e-mail. This element is optional. This is a read/write element.
IsReadReceiptRequested	xs:boolean	The <IsReadReceiptRequested> element gets or sets a Boolean value that specifies whether the sender of a message requests a read receipt. This element is optional. This is a read/write element.
IsDeliveryReceiptRequested	xs:boolean	The <IsDeliveryReceiptRequested> element gets or sets a Boolean value that specifies whether the delivery receipt is serialized into the SOAP request. This element is required when the <IsDeliveryReceiptRequested> element is specified. This is a read/write element.
ConversationIndex	xs:base64Binary	The <ConversationIndex> element gets or sets the identifier to the thread to which the message belongs. This element is optional. This is a read/write element.

Element	Type	Description
ConversationTopic	xs:string	The <ConversationTopic> element gets or sets the conversation identifier. This element is optional. This is a read/write element.
From	t:SingleRecipientType	The <From> element gets or sets the addressee from whom the message was sent. This element is optional. This is a read/write element.
InternetMessageId	xs:string	The <InternetMessageId> element gets or sets the Internet message identifier for the message. This element is optional. This is a read/write element.
IsRead	xs:boolean	The <IsRead> element gets a Boolean value that specifies whether the message has been read. This element is read-only.
IsResponseRequested	xs:boolean	The <IsResponseRequested> element gets or sets a Boolean value that specifies whether a response to an e-mail is requested. This element is optional. This is a read/write element.
References	xs:string	The <References> element gets or sets a string that represents the Usenet header that is used to correlate replies with their original message. This element is optional. This is a read/write element.
ReplyTo	t:ArrayOfRecipientsType	The <ReplyTo> element gets or sets a collection of addresses to which replies should be sent. This element is optional. This is a read/write element.
ReceivedBy	t:SingleRecipientType	The <ReceivedBy> element identifies the delegate in a delegate access scenario.
ReceivedRepresenting	t:SingleRecipientType	The <ReceivedRepresenting> element identifies the principal in a delegate access scenario.

2.2.5 Simple Types

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

Simple Type	Description
t:MessageDispositionType	The MessageDispositionType enumeration specifies how a message item is handled after it is created or updated.

2.2.5.1 t:MessageDispositionType Simple Type

The **MessageDispositionType** simple type enumeration specifies how a message item is handled after it is created or updated.

```
<xs:simpleType name="MessageDispositionType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="SaveOnly"
    />
    <xs:enumeration
      value="SendOnly"
    />
    <xs:enumeration
      value="SendAndSaveCopy"
    />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

The following values are defined by the **MessageDispositionType** simple type:

Value	Description
SaveOnly	When used in the <CreateItemType> element (section 3.1.4.2.1.2), the e-mail message item is saved in the folder that is specified by the TargetFolderIdType complex type (section 2.2.4.14). Messages can be sent later by using the SendItem operation (section 3.1.4.6.2.1) on an ExchangeServiceBinding object. In this case, an item identifier is returned.
SendOnly	When used in the <CreateItemType> element, the e-mail message item is sent but no copy is saved. In this case, an item identifier is not returned.
SendAndSaveCopy	When used in the <CreateItemType> element, the e-mail message item is sent and a copy is saved in the TargetFolderIdType complex type. In this case, an item identifier is not returned.

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

The Core Items Web Service protocol defines a single port type.

Operation	Description
GetItem	Defines a request to get a message item from a mailbox on the server.
CreateItem	Defines a request to create a message item on the server.
DeleteItem	Defines a request to delete a message item from a mailbox on the server.
UpdateItem	Defines a request to update a message item in a mailbox.
MoveItem	Defines a request to move a message item on the server.
CopyItem	Defines a request to copy a message item in a mailbox on the server.
SendItem	Defines a request to send a message item element.

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 specifies the operations listed in the following table.

Operation	Description
GetItem	Gets items from the server.
CreateItem	Creates items on the server.
DeleteItem	Deletes items from the server.
UpdateItem	Updates items on the server.

Operation	Description
SendItem	Sends items to the server.
MoveItem	Moves items on the server.
CopyItem	Copies items on the server.

3.1.4.1 CreateItem

This protocol uses the **CreateItem** operation to create e-mail messages.

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

Request

Message Format	Description
tns:CreateItemSoapIn	Specifies the SOAP message that defines the message item to create. The Items child element of the CreateItem child element that specifies the XML request MUST contain one or more <MessageType> elements. All other elements MUST be empty.

Response

Message Format	Description
tns:CreateItemSoapOut	Specifies the [SOAP] message returned by the server in response.

3.1.4.2 GetItem

This protocol uses the **GetItem** operation to access information about e-mail messages.

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

Request

Message Format	Description
tns:GetItemSoapIn	Specifies the SOAP message that defines the message item to get. The <Items> child element of the <GetItem> element ([MS-OXWSCORE] section 3.1.4.4.2.1) that specifies the XML request MUST contain the ItemResponseShapeType complex type ([MS-OXWSCDATA] section 2.2.3.35) and the ItemIdType complex type ([MS-OXWSCORE] section 2.2.3.13).

Response

Message Format	Description
tns:GetItemSoapOut	Specifies the SOAP message that is returned by the server in response. The server returns an <ItemResponseShapeType> ([MS-OXWSCDATA] section 2.2.3.35) element that contains properties that are associated with the message item.

The **GetItem** operation request MUST include the elements listed in the following table.

Element	Description
<ItemShape>	Specifies a set of properties to be returned. The child elements for this element are listed in the following table.
<ItemIds>	Contains the unique identities of items. This element includes the Id , which identifies a specific item in the store, and the ChangeKey attribute, which identifies a specific version of an item.

The child elements of the <ItemShape> element are listed in the following table.

Element	Description
<BaseShape>	Identifies the basic configuration of properties to return in an item response. Set to IdOnly to return only the item ID or set to AllProperties to return all the properties used by the server to construct a message. This element MUST be present.
<IncludeMimeContent>	Indicates whether MIME content is included in the returned message or attachment.
<BodyType>	Indicates whether a message body is returned as HTML.
<FilterHtmlContent>	Indicates whether to filter unsafe HTML content from the message or attachment .
<AdditionalProperties>	Identifies additional item properties to return in a response.

3.1.4.3 UpdateItem

This protocol uses the **UpdateItem** operation to update e-mail message properties in the server store.

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

Request

Message Format	Description
tns:UpdateItemSoapIn	Specifies the SOAP message that defines the message item to update.

Response

Message Format	Description
tns:UpdateItemSoapOut	Specifies the SOAP message that is returned by the server in response.

The **UpdateItem** operation modifies a message that is already created and sent. The **UpdateItem** operation request MAY contain the **MessageDispositionType** simple type (section [2.2.5.1](#)) and MUST contain the **ConflictResolutionType** simple type (section [3.1.4.7.1.1](#)).

3.1.4.4 DeleteItem

This protocol uses the **DeleteItem** operation to delete e-mail messages from the server store

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

Request

Message Format	Description
tns>DeleteItemSoapIn	Specifies the SOAP message that defines the message item to delete. The <Items> child element of the <DeleteItem> element ([MS-OXWSCORE] section 3.1.4.3.2.1) that specifies the XML request MUST contain the TargetFolderIdType complex type ([MS-OXWSFOLD] section 2.2.4.14) and the ItemIdType complex type ([MS-OXWSCORE] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns>DeleteItemSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.5 MoveItem

This protocol uses the **MoveItem** operation to move one or more e-mail messages to a single destination folder.

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

Request

Message Format	Description
tns:MoveItemSoapIn	Specifies the SOAP message that defines the message item to move. The <MoveItem> element ([MS-OXWSCORE] section 3.1.4.5.2.1) that specifies the XML request MUST contain the TargetFolderIdType complex type ([MS-OXWSFOLD] section 2.2.4.14) and the ItemIdType complex type ([MS-

Message Format	Description
	OXWScore section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:MoveItemSoapOut Message	Specifies the SOAP message that is returned by the server in response.

3.1.4.6 CopyItem

This protocol uses the **CopyItem** operation to copy e-mail messages and puts the messages in a different folder.

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

Request

Message Format	Description
tns:CopyItemSoapIn	Specifies the SOAP message that defines the message item to copy. The <GetItem> element ([MS-OXWScore] section 3.1.4.4.2.1) that specifies the XML request MUST contain the TargetFolderIdType complex type ([MS-OXWSFOLD] section 2.2.4.14) and the ItemIdType complex type ([MS-OXWScore] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:CopyItemSoapOut	Specifies the SOAP message that is returned by the server in response.

3.1.4.7 SendItem

This protocol uses the **SendItem** operation to send e-mail messages that are located in the server store.

```
<wsdl:operation name="SendItem ">
  <wsdl:input message="tns:SendItemSoapIn" />
  <wsdl:output message="tns:SendItemSoapOut" />
</wsdl:operation>
```

Request

Message Format	Description
tns:SendItemSoapIn	Specifies the SOAP message that defines the message item to send. The Items child element of the <SendItem> element ([MS-OXWSCORE] section 3.1.4.6) that specifies the XML request MUST contain the TargetFolderIdType complex type ([MS-OXWSFOLD] section 2.2.4.14) and the ItemIdType complex type ([MS-OXWSCORE] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:SendItemSoapOut	Specifies the SOAP message that is returned by the server in response.

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 for the operations that this protocol uses.

4.1 Create Message Example

The following is an example of a **CreateItem** operation (section [3.1.4.1](#)) that is used to create a message to User1 and User2 at Contoso.com. This example creates a mail and has only the Subject and Body properties completed. The e-mail message can be set with many more properties, such as attachments, Bcc recipients, categories, sender, and item class. This example shows how to send an e-mail message and save a copy of the message in the default Sent Items folder by using the `SendAndSaveCopy()` method.

The client constructs the request XML and sends it to the server. The newly created message is sent to the server.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m:CreateItem MessageDisposition="SendAndSaveCopy">
      <m:SavedItemFolderId>
        <t:DistinguishedFolderId Id="sentitems" />
      </m:SavedItemFolderId>
      <m:Items>
        <t:Message>
          <t:Subject>Interesting</t:Subject>
          <t:Body BodyType="HTML">The merger is finalized.</t:Body>
          <t:ToRecipients>
            <t:Mailbox>
              <t:EmailAddress>User1@Contoso.com</t:EmailAddress>
            </t:Mailbox>
            <t:Mailbox>
              <t:EmailAddress>User2@Contoso.com</t:EmailAddress>
            </t:Mailbox>
          </t:ToRecipients>
        </t:Message>
      </m:Items>
    </m:CreateItem>
  </soap:Body>
</soap:Envelope>
```

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

```
<?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:CreateItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:CreateItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Items />
        </m:CreateItemResponseMessage>
      </m:ResponseMessages>
    </m:CreateItemResponse>
  </s:Body>
</s:Envelope>

```

4.2 Get Message Example

The following is an example of a [GetItem](#) operation with the [m:GetItemType](#) complex type. This example gets a message from the server store. The message must already be identified with its ItemId Id and ChangeKey.

The client constructs the request XML and sends it to the server. The ItemId Id and ChangeKey attributes have 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:GetItem>
      <m:ItemShape>
        <t:BaseShape>IdOnly</t:BaseShape>
        <t:AdditionalProperties>
          <t:FieldURI FieldURI="item:Body" />
        </t:AdditionalProperties>
      </m:ItemShape>
      <m:ItemIds>
        <t:ItemId Id="AAMkAGY4YzQw" ChangeKey="CQAAAByAAA " />
      </m:ItemIds>
    </m:GetItem>
  </soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client. The ItemId Id and ChangeKey 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:GetItemResponse
        xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
        xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
        <m:ResponseMessages>
          <m:GetItemResponseMessage ResponseClass="Success">
            <m:ResponseCode>NoError</m:ResponseCode>
            <m:Items>
              <t:Message>
                <t:ItemId Id="AAMkAGY4YzQw" ChangeKey="CQAAABYAAA" />
                <t:Body BodyType="HTML">&lt;meta http-equiv="Content-Type"
content="text/html;
                charset=utf-8"&gt;The merger is finalized.</t:Body>
              </t:Message>
            </m:Items>
          </m:GetItemResponseMessage>
        </m:ResponseMessages>
      </m:GetItemResponse>
    </s:Body>
  </s:Envelope>

```

4.3 Update Message Example

The following is an example of an **UpdateItem** operation (section [3.1.4.3](#)) with the **UpdateItemType** complex type (section [3.1.4.7.2.2](#)). This example updates the subject of an existing message.

The client constructs the request XML and sends it to the server. The **ItemId** and **ChangeKey** attributes have 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:UpdateItem MessageDisposition="SaveOnly" ConflictResolution="AlwaysOverwrite">
      <m:ItemChanges>
        <t:ItemChange>
          <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />

```

```

    <t:Updates>
      <t:SetItemField>
        <t:FieldURI FieldURI="item:Subject" />
        <t:Message>
          <t:Subject>Modified and updated mail</t:Subject>
        </t:Message>
      </t:SetItemField>
    </t:Updates>
  </t:ItemChange>
</m:ItemChanges>
</m:UpdateItem>
</soap:Body>
</soap:Envelope>

```

The server constructs the response XML and sends it to the client. The **ItemId** and **ChangeKey** 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:UpdateItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:UpdateItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Items>
            <t:Message>
              <t:ItemId Id="AAMkAGIwODEy=" ChangeKey="CQAAABYAAA" />
            </t:Message>
          </m:Items>
          <m:ConflictResults>
            <t:Count>0</t:Count>
          </m:ConflictResults>
        </m:UpdateItemResponseMessage>
      </m:ResponseMessages>
    </m:UpdateItemResponse>
  </s:Body>
</s:Envelope>

```

4.4 Delete Message Example

The following is an example of a [DeleteItem](#) operation with the [DeleteItemType](#) complex type. This example deletes the identified message from the server store.

The client constructs the request XML and sends it to the server. The newly created message is sent to the server. The ItemId Id attribute is 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:DeleteItem DeleteType="HardDelete">
      <m:ItemIds>
        <t:ItemId Id="AAMkAGY4YzQw" />
      </m:ItemIds>
    </m:DeleteItem>
  </soap:Body>
</soap:Envelope>
```

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

```
<?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:DeleteItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:DeleteItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
        </m:DeleteItemResponseMessage>
      </m:ResponseMessages>
    </m:DeleteItemResponse>
  </s:Body>
</s:Envelope>
```

4.5 Move Message Example

The following is an example of a **MoveItem** operation (section [3.1.4.5](#)) with a **MoveItemType** complex type (section [3.1.4.5.1.2](#)). This example moves an identified message to the **Junk E-mail folder**.

The client constructs the request XML and sends it to the server. The **ItemId** 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:MoveItem>
      <m:ToFolderId>
        <t:DistinguishedFolderId Id="junkemail" />
      </m:ToFolderId>
      <m:ItemIds>
        <t:ItemId Id="AAMkAGIw " />
      </m:ItemIds>
    </m:MoveItem>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **ItemId** and **ChangeKey** 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:MoveItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:MoveItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Items>
            <t:Message>
              <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />
            </t:Message>
          </m:Items>
        </m:MoveItemResponseMessage>
      </m:ResponseMessages>
    </m:MoveItemResponse>
  </s:Body>
</s:Envelope>
```


4.6 Copy Message Example

The following is an example of a **CopyItem** operation (section [3.1.4.6](#)) with a **CopyItemType** complex type (section [3.1.4.1.1.2](#)). This example copies an identified message to the Junk E-mail folder.

The client constructs the request XML and sends it to the server. The **ItemId** and **ChangeKey** attributes have 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:CopyItem>
      <m:ToFolderId>
        <t:DistinguishedFolderId Id="junkemail" />
      </m:ToFolderId>
      <m:ItemIds>
        <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />
      </m:ItemIds>
    </m:CopyItem>
  </soap:Body>
</soap:Envelope>
```

The server constructs the response XML and sends it to the client. The **ItemId** and **ChangeKey** 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:CopyItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
      <m:ResponseMessages>
        <m:CopyItemResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:Items>
```

```

        <t:Message>
          <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />
        </t:Message>
      </m:Items>
    </m:CopyItemResponseMessage>
  </m:ResponseMessages>
</m:CopyItemResponse>
</s:Body>
</s:Envelope>

```

4.7 Send Message Example

The following is an example of a **SendItem** operation (section [3.1.4.7](#)) with the **SendItemType** complex type (section [2.2.3.3](#)). This example sends an identified message.

The client constructs the request XML and sends it to the server. The **ItemId** and **ChangeKey** attributes have 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:SendItem SaveItemToFolder="false">
      <m:ItemIds>
        <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />
      </m:ItemIds>
    </m:SendItem>
  </soap:Body>
</soap:Envelope>
</EwsLogEntry>

```

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

```

<?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:SendItemResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">

```

```
<m:ResponseMessages>
  <m:SendItemResponseMessage ResponseClass="Success">
    <m:ResponseCode>NoError</m:ResponseCode>
  </m:SendItemResponseMessage>
</m:ResponseMessages>
</m:SendItemResponse>
</s:Body>
</s:Envelope>
</EwsLogEntry>
```

5 Security

5.1 Security Considerations for Implementers

The E-Mail Message Types Web Service does not use additional security mechanisms.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The following XML files are required to implement the functionality described in this specification. The contents of each file are contained in this section.

Section	Protocol Filename	Description
WSDL	MS-OXWSMSG.wsdl	Contains the WSDL for the implementation of this protocol.
Types Schema	MS-OXWSMSG-types.xsd	Contains the XML schema type definitions 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 or imported into the MS-OXWSMSG-types.xsd schema MUST be placed in the common folder with the files named in the table.

6.1 WSDL

This section contains the contents of the MS-OXWSMSG.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:operation name="GetAttachment" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:input message="tns:GetAttachmentSoapIn"/>
    </wsdl:operation>
  </wsdl:portType>

```

```

        <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-OXWSMSG-types.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSMSG-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
[MS-OXWSCORE], section 6.2	MS-OXWSCORE -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-OXWSCORE-types.xsd"/>
  <xs:complexType name="MessageType">
    <xs:complexContent>
      <xs:extension base="t:ItemType">
        <xs:sequence>
          <xs:element name="Sender" type="t:SingleRecipientType"
minOccurs="0"/>
          <xs:element name="ToRecipients" type="t:ArrayOfRecipientsType"
minOccurs="0"/>
          <xs:element name="CcRecipients" type="t:ArrayOfRecipientsType"
minOccurs="0"/>
          <xs:element name="BccRecipients" type="t:ArrayOfRecipientsType"
minOccurs="0"/>
          <xs:element name="IsReadReceiptRequested" type="xs:boolean"
minOccurs="0"/>
          <xs:element name="IsDeliveryReceiptRequested" type="xs:boolean"
minOccurs="0"/>
          <xs:element name="ConversationIndex" type="xs:base64Binary"
minOccurs="0"/>
          <xs:element name="ConversationTopic" type="xs:string"
minOccurs="0"/>
          <xs:element name="From" type="t:SingleRecipientType" minOccurs="0"/>
          <xs:element name="InternetMessageId" type="xs:string"
minOccurs="0"/>
          <xs:element name="IsRead" type="xs:boolean" minOccurs="0"/>
          <xs:element name="IsResponseRequested" type="xs:boolean"
minOccurs="0"/>
          <xs:element name="References" type="xs:string" minOccurs="0"/>
          <xs:element name="ReplyTo" type="t:ArrayOfRecipientsType"
minOccurs="0"/>
          <xs:element name="ReceivedBy" type="t:SingleRecipientType"
minOccurs="0"/>
          <xs:element name="ReceivedRepresenting" type="t:SingleRecipientType"
minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:simpleType name="MessageDispositionType">
```



```
<xs:restriction base="xs:string">
  <xs:enumeration value="SaveOnly"/>
  <xs:enumeration value="SendOnly"/>
  <xs:enumeration value="SendAndSaveCopy"/>
</xs:restriction>
</xs:simpleType>
</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-OXWSMSG] 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.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
Z Appendix B: Product Behavior	57836 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](#) 35

F

[Full WSDL](#) 29

G

[Glossary](#) 5

I

[Introduction](#) 5

M

Messages
[overview](#) 8
[syntax](#) 8
[transport](#) 8

O

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

P

[Preconditions](#) 6
[Prerequisites](#) 6
[Product Behavior](#) 34

R

References
[informative](#) 6
[normative](#) 5
[Relationship to other protocols](#) 6

S

Security
implementer considerations ([section 5.1](#) 28,
[section 5.2](#) 28)
[overview](#) 28

Server
[abstract data model](#) 13
[initialization](#) 13
[local events](#) 18
[message processing](#) 13
[overview](#) 13
[sequencing rules](#) 13

[timer events](#) 18

[timers](#) 13

[Standards assignments](#) 7

T

[Tracking changes](#) 35

V

[Vendor-extensible fields](#) 7

[Versioning](#) 7