

# [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 [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.

**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.

PRELIMINARY

# Table of Contents

<b>1 Introduction .....</b>	<b>5</b>
1.1 Glossary.....	5
1.2 References.....	5
1.2.1 Normative References .....	5
1.2.2 Informative References .....	6
1.3 Protocol Overview .....	6
1.4 Relationship to Other Protocols.....	6
1.5 Prerequisites/Preconditions.....	6
1.6 Applicability Statement.....	6
1.7 Versioning and Capability Negotiation.....	7
1.8 Vendor-Extensible Fields .....	7
1.9 Standards Assignments .....	7
<b>2 Messages .....</b>	<b>8</b>
2.1 Transport.....	8
2.2 Common Message Syntax.....	8
2.2.1 Namespaces.....	8
2.2.2 Simple Types .....	8
2.2.2.1 t:MessageDispositionType Simple Type.....	8
2.2.3 Complex Types.....	9
2.2.3.1 t:MessageType Complex Type .....	9
2.2.4 Elements.....	12
2.2.5 Attributes.....	12
2.2.6 Groups.....	12
2.2.7 Attribute Groups .....	12
2.2.8 Message Syntax.....	12
<b>3 Protocol Details.....</b>	<b>13</b>
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.....	15
3.1.4.5 MoveItem .....	16
3.1.4.6 CopyItem.....	16
3.1.4.7 SendItem.....	16
3.1.5 Timer Events.....	17
3.1.6 Other Local Events .....	17
3.2 Client Details.....	17
3.2.1 Abstract Data Model.....	17
3.2.2 Initialization .....	17
3.2.3 Initialization .....	17
3.2.4 Message Processing Events and Sequencing .....	17
3.2.5 Timer Events.....	17
3.2.6 Other Local Events .....	18

<b>4 Protocol Examples .....</b>	<b>19</b>
<b>5 Security.....</b>	<b>20</b>
5.1 Security Considerations for Implementers.....	20
5.2 Index of Security Parameters .....	20
<b>6 Appendix A: Full WSDL .....</b>	<b>21</b>
6.1 Types Schema.....	21
6.2 WSDL.....	22
<b>7 Appendix B: Product Behavior .....</b>	<b>26</b>
<b>8 Change Tracking .....</b>	<b>27</b>
<b>9 Index.....</b>	<b>28</b>

PRELIMINARY

# 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 [\[MS-OXGLOS\]](#):

**certificate**  
**SOAP body**  
**SOAP fault**  
**SOAP header**  
**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-OXWSCORE] Microsoft Corporation, "[Core 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>.

[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/>.

## 1.2.2 Informative References

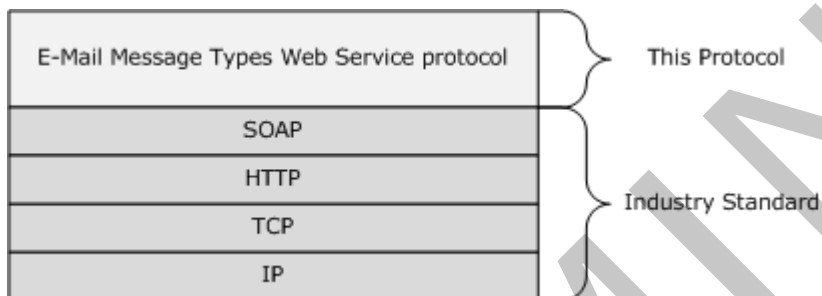
None.

## 1.3 Protocol Overview

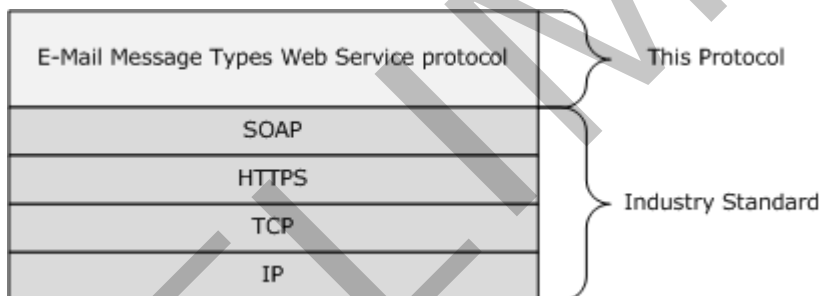
The E-Mail Message Types Web Service protocol provides clients with the ability to create, update, and delete e-mail items on the server. Clients create e-mail items using the CreateItem operation, or get the properties of an existing task item using the GetItem operation. E-mails can also be sent, updated, 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: Figure 1:SOAP over HTTP**



**Figure 2: 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	<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-OXWSMSG]
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-OXWSMSG]
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>
t	<a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a>	[MS-OXWSMSG]

#### 2.2.2 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
<a href="#">t:MessageDispositionType</a>	The <b>MessageDispositionType</b> enumeration specifies how a message item is handled after it is created or updated.

##### 2.2.2.1 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, the e-mail message item is saved in the folder that is specified by the <a href="#">TargetFolderIdType</a> property. Messages can be sent later by using the <a href="#">SendItem</a> operation 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 <b>TargetFolderIdType</b> property. In this case, an item identifier is not returned.

## 2.2.3 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
<a href="#">t:MessageType</a>	Represents a server e-mail message in a mailbox.

### 2.2.3.1 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	<a href="#">t:SingleRecipientType</a>	The <Sender> element gets or sets the sender of a message. This element is optional. This is a read/write element.
ToRecipients	<a href="#">t:ArrayOfRecipientsType</a>	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	<b>t:ArrayOfRecipientsType</b>	The <CcRecipients> element gets or sets a collection of recipients that receive a carbon copy (Bcc) of an e-mail. This element is optional. This is a read/write element.
BccRecipients	<b>t:ArrayOfRecipientsType</b>	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	<b>xs:boolean</b>	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	<b>xs:boolean</b>	The <IsDeliveryReceiptRequestedSpecified> element gets or sets a Boolean value that specifies whether the <IsDeliveryReceiptRequested> element is serialized into the Simple Object Access protocol (SOAP) request. This element is required when the <IsDeliveryReceiptRequested> element is specified. This is a read/write element.
ConversationIndex	<b>xs:base64Binary</b>	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.
ConversationTopic	<b>xs:string</b>	The <ConversationTopic> element gets or sets the conversation identifier. This element is optional. This is a read/write element.
From	<b>t:SingleRecipientType</b>	The <From> element gets or sets the addressee from whom the message was sent. This element is optional. This is a read/write element.

Element	Type	Description
InternetMessageId	<b>xs:string</b>	The <InternetMessageId> element gets or sets the Internet message identifier for the message. This element is optional. This is a read/write element.
IsRead	<b>xs:boolean</b>	The <IsRead> element gets a Boolean value that specifies whether the message has been read. This element is read-only.
IsResponseRequested	<b>xs:boolean</b>	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	<b>xs:string</b>	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	<b>t:ArrayOfRecipientsType</b>	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	<b>t:SingleRecipientType</b>	The <ReceivedBy> element identifies the delegate in a delegate access scenario.
ReceivedRepresenting	<b>t:SingleRecipientType</b>	The <ReceivedRepresenting> element identifies the principal in a delegate access scenario.

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

Operation	Description
<a href="#">GetItem</a>	Defines a request to get a message item from a mailbox on the server.
<a href="#">CreateItem</a>	Defines a request to create a message item on the server.
<a href="#">DeleteItem</a>	Defines a request to delete a message item from a mailbox on the server.
<a href="#">UpdateItem</a>	Defines a request to update a message item in a mailbox.
<a href="#">MoveItem</a>	Defines a request to move a message item on the server.
<a href="#">CopyItem</a>	Defines a request to copy a message item in a mailbox on the server.
<a href="#">SendItem</a>	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
<a href="#">GetItem</a>	Gets items from the server.
<a href="#">CreateItem</a>	Creates items on the server.
<a href="#">DeleteItem</a>	Deletes items from the server.
<a href="#">UpdateItem</a>	Updates items on the server.

Operation	Description
<a href="#">SendItem</a>	Sends items to the server.
<a href="#">MoveItem</a>	Moves items on the server.
<a href="#">CopyItem</a>	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
<a href="#">tns:CreateItemSoapIn</a>	Specifies the SOAP message that defines the message item to create. The <a href="#">Items</a> child element of the <b>CreateItem</b> child element that specifies the XML request MUST contain one or more <a href="#">t:MessageType</a> elements. All other elements MUST be empty.

Response

Message Format	Description
<a href="#">tns:CreateItemSoapOut</a>	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
<a href="#">tns:GetItemSoapIn</a>	Specifies the SOAP message that defines the message item to get. The <a href="#">Items</a> child element of the <a href="#">GetItem</a> child element that specifies the XML request MUST contain the <a href="#">t:ItemResponseShapeType</a> and <a href="#">t:ItemIdType</a> elements.

Response

Message Format	Description
<a href="#">tns:GetItemSoapOut</a>	Specifies the [SOAP] message returned by the server in response. The server returns a <a href="#">t:ItemResponseShapeType</a> element that contains properties associated with the message item.

### 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
<a href="#">tns:UpdateItemSoapIn</a>	Specifies the SOAP message that defines the calendar item to update.

Response

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

### 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
<a href="#">tns&gt;DeleteItemSoapIn</a>	Specifies the SOAP message that defines the message item to delete. The <a href="#">Items</a> child element of the <a href="#">DeleteItem</a> child element that specifies the XML request MUST contain one or more <a href="#">t:ItemIdType</a> elements. All other elements MUST be empty.

Response

Message Format	Description
<a href="#">tns&gt;DeleteItemSoapOut</a>	Specifies the [SOAP] message 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
<a href="#">tns:MoveItemSoapIn</a>	Specifies the SOAP message that defines the message item to move. The <a href="#">Items</a> child element of the <a href="#">MoveItem</a> child element that specifies the XML request MUST contain the <a href="#">t:TargetFolderIdType</a> and <a href="#">t:ItemIdType</a> elements. All other elements MUST be empty.

Response

Message Format	Description
<a href="#">tns:MoveItemSoapOut</a>	Specifies the SOAP message 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
<a href="#">tns:CopyItemSoapIn</a>	Specifies the SOAP message that defines the message item to copy. The <a href="#">Items</a> child element of the <a href="#">CopyItem</a> child element that specifies the XML request MUST contain the <a href="#">t:TargetFolderIdType</a> and <a href="#">t:ItemIdType</a> elements. All other elements MUST be empty.

Response

Message Format	Description
<a href="#">tns:CopyItemSoapOut</a>	Specifies the SOAP message 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" />

```

#### Request

Message Format	Description
<a href="#">tns:SendItemSoapIn</a>	Specifies the SOAP message that defines the message item to send. The <a href="#">Items</a> child element of the <a href="#">SendItem</a> child element that specifies the XML request MUST contain the <a href="#">t:TargetFolderIdType</a> and <a href="#">t:ItemIdType</a> elements. All other elements MUST be empty.

#### Response

Message Format	Description
<a href="#">tns:SendItemSoapOut</a>	Specifies the SOAP message returned by the server in 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 Initialization

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.

PRELIMINARY

## 4 Protocol Examples

None.

PRELIMINARY

## 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.

PRELIMINARY

## 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
<a href="#">WSDL</a>	MS-OXWSMSG.wsdl	Contains the <b>WSDL</b> for the implementation of this protocol.
<a href="#">Types Schema</a>	MS-OXWSMSG-types.xsd	Contains the <b>XML schema</b> 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 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 <a href="#">6.2</a>	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:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

        <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>

```

## 6.2 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: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>
```

PRELIMINARY

## 7 Appendix B: Product Behavior

The information in this specification is applicable to the following product versions:

- 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 the product does not follow the prescription.

## 8 Change Tracking

This section will report content and/or editorial changes, beginning with the next release.

PRELIMINARY

## 9 Index

### A

[Applicability](#)

### C

[Capability negotiation](#)

[Change tracking](#)

Client

[abstract data model](#)

[initialization](#)

[local events](#)

[message processing](#)

[overview](#)

[sequencing rules](#)

[timer events](#)

[timers](#)

### F

[Full WSDL](#)

### G

[Glossary](#)

### I

[Introduction](#)

### M

Messages

[overview](#)

[syntax](#)

[transport](#)

### O

[Overview](#)

### P

[Preconditions](#)

[Prerequisites](#)

[Product Behavior](#)

### R

References

[informative](#)

[normative](#)

[Relationship to other protocols](#)

### S

Security

implementer considerations ([section 5.1](#), [section 5.2](#))

[overview](#)

Server

[abstract data model](#)

[initialization](#)

[local events](#)

[message processing](#)

[overview](#)

[sequencing rules](#)

[timer events](#)

[timers](#)

[Standards assignments](#)

### T

[Tracking changes](#)

### V

[Vendor-extensible fields](#)

[Versioning](#)