[MS-OXWSMSG]:

Email Message Types Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	2.0.0	Major	Updated and revised the technical content.
2/10/2010	3.0.0	Major	Updated and revised the technical content.
5/5/2010	3.1.0	Minor	Updated the technical content.
8/4/2010	4.0	Major	Significantly changed the technical content.
11/3/2010	5.0	Major	Significantly changed the technical content.
3/18/2011	6.0	Major	Significantly changed the technical content.
8/5/2011	6.1	Minor	Clarified the meaning of the technical content.
10/7/2011	6.1	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	7.0	Major	Significantly changed the technical content.
4/27/2012	7.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	7.1	Minor	Clarified the meaning of the technical content.
2/11/2013	7.2	Minor	Clarified the meaning of the technical content.
7/26/2013	8.0	Major	Significantly changed the technical content.
11/18/2013	8.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	8.0	None	No changes to the meaning, language, or formatting of the technical content.
5/26/2015	9.0	Major	Significantly changed the technical content.
9/14/2015	10.0	Major	Significantly changed the technical content.
6/13/2016	11.0	Major	Significantly changed the technical content.
9/14/2016	11.0	None	No changes to the meaning, language, or formatting of the technical content.
1/12/2017	11.1	Minor	Clarified the meaning of the technical content.
9/19/2017	12.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
7/24/2018	13.0	Major	Significantly changed the technical content.
10/1/2018	14.0	Major	Significantly changed the technical content.
11/19/2019	14.1	Minor	Clarified the meaning of the technical content.
2/15/2022	14.1	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Email Message Types Web Service Protocol is used to create, get, update, delete, move, copy, and send email messages in a mailbox.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

delegate: A user or resource that has permissions to act on behalf of another user or resource.

delegate access: The access that is granted by a delegator to a delegate and is used by the delegate to access the delegator's account.

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

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

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

Junk Email folder: A special folder that is the default location for Message objects that are determined to be junk email by a Junk Email rule.

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

message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

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

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses **XML** technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

web server: A server computer that hosts websites and responds to requests from applications.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network

service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a **WSDL operation** [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL operation: A single action or function of a web service. The execution of a WSDL operation typically requires the exchange of messages between the service requestor and the service provider.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [XML1.0].

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

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

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

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

[MS-OXWSFOLD] Microsoft Corporation, "Folders and Folder Permissions Web Service Protocol".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, https://www.rfc-editor.org/rfc/rfc2119.html

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, https://www.rfc-editor.org/info/rfc2616

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, https://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, https://www.w3.org/TR/2001/NOTE-wsdl-20010315

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

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

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

1.2.2 Informative References

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

[MS-OXWSSRCH] Microsoft Corporation, "Mailbox Search Web Service Protocol".

1.3 Overview

The Email Message Types Web Service Protocol provides clients with the ability to create, get, update, delete, move, copy, and send email messages on the server. Clients create email messages by using the **CreateItem** operation, or they get the properties of an existing task item by using the **GetItem** operation. Email messages can also be sent, updated, deleted, moved, or copied on the server by using the **SendItem**, **UpdateItem**, **DeleteItem**, **MoveItem**, and **CopyItem** operations, respectively.

1.4 Relationship to Other Protocols

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

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

This protocol uses SOAP over **HTTP**, as described in [RFC2616], and SOAP over **HTTPS**, as described in [RFC2818], as shown in the following figure.

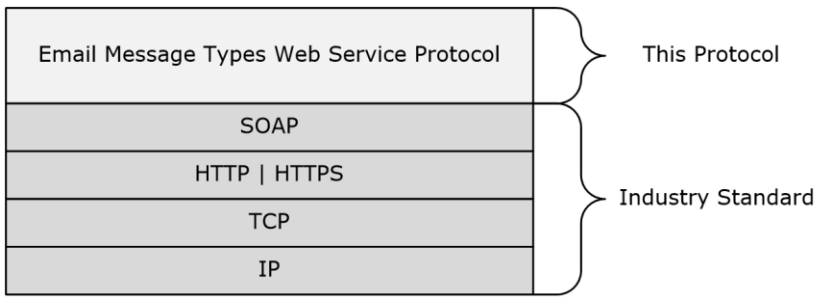

Figure 1: This protocol in relation to other protocols

This protocol uses the email message identifier returned by the Mailbox Search Web Service Protocol, as described in [MS-OXWSSRCH], to retrieve information from the **message store**.

This protocol uses the **CopyItem**, **CreateItem**, **DeleteItem**, **GetItem**, **MoveItem**, **SendItem**, and **UpdateItem** operations of the Core Items Web Service Protocol, as described in [MS-OXWSCORE], to manipulate an email message.

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

The **endpoint URL** that is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as specified by [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as specified by [MS-OXDSCLI], is required to form the **HTTP** request to the **web server** that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP Web requests that target this protocol.

To access this protocol, all callers are authenticated. This protocol relies on the web server that hosts the application to perform authentication.

1.6 Applicability Statement

This protocol is applicable to environments that copy, create, delete, get, send, move, or update email messages by using Exchange Web Services.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with **SOAP** 1.1, as described in section 2.1.
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **Web Services Description Language (WSDL)** version of the request is identified by using the **t:RequestServerVersion** element, as described in [MS-OXWSCDATA] section 2.2.3.9, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [MS-OXWSCDATA] section 2.2.3.10.
- **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 described in section 3.1.4.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The **WSDL** in this specification provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, or **present**.

2.1 Transport

This protocol uses **SOAP** 1.1. For details, see [SOAP1.1].

This protocol relies on the **web server** that hosts the application to perform authentication. The protocol MUST support SOAP over **HTTP**, as specified in [RFC2616]. The protocol SHOULD use secure communications via **HTTPS**, as defined in [RFC2818].

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and **WSDL**, as specified in [WSDL].

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML 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	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	

2.2.2 Messages

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

2.2.3 Elements

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

2.2.4 Complex Types

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

Complex type name	Description
ApprovalRequestDataType (section 2.2.4.1)	Represents an approval request message.<1>
ArrayOfVotingOptionDataType (section 2.2.4.2)	Specifies an array of voting options.<2>
MessageType (section 2.2.4.3)	Represents a server email message in a user's **mailbox**.
ReminderMessageDataType (section 2.2.4.4)	Specifies a reminder message.<3>
VotingInformationType (section 2.2.4.5)	Specifies voting information.<4>
VotingOptionDataType (section 2.2.4.6)	Specifies the voting option on a message.<5>

2.2.4.1 t:ApprovalRequestDataType Complex Type

The **ApprovalRequestDataType** complex type represents an approval request message.<6>

```
<xs:complexType name="ApprovalRequestDataType">
  <xs:sequence>
    <xs:element name="IsUndecidedApprovalRequest" type="xs:boolean" minOccurs="0"/>
    <xs:element name="ApprovalDecision" type="xs:int" minOccurs="0"/>
    <xs:element name="ApprovalDecisionMaker" type="xs:string" minOccurs="0"/>
    xs:element name="ApprovalDecisionTime" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

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

Element	Type	Description
IsUndecidedApprovalRequest	**xs:boolean** ([XMLSCHEMA2])	A value that indicates whether this approval request is awaiting a moderator to approve or reject the request.
ApprovalDecision	**xs:int** ([XMLSCHEMA2])	The approval decision on the approval request message.
ApprovalDecisionMaker	**xs:string** ([XMLSCHEMA2])	The display name of the moderator who approved or rejected the request.
ApprovalDecisionTime	**xs:dateTime** ([XMLSCHEMA2])	The time at which a moderator approved or rejected the request.

2.2.4.2 t:ArrayOfVotingOptionDataType Complex Type

The **ArrayOfVotingOptionDataType** complex type specifies an array of voting options.<7>

```
<xs:complexType name="ArrayOfVotingOptionDataType">
  <xs:sequence>
    <xs:element name="VotingOptionData" type="t:VotingOptionDataType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

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

Element	Type	Description
VotingOptionData	**t:VotingOptionDataType** (section 2.2.4.6)	The voting options on a message.

2.2.4.3 t:MessageType Complex Type

The **MessageType** complex type represents a server email message in a mailbox. The **MessageType** complex type extends the **ItemType** complex type ([MS-OXWSCORE] section 2.2.4.24).

```
<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:element name="ApprovalRequestData"
            type="t:ApprovalRequestDataType"
            minOccurs="0"
           />
          <xs:element name="VotingInformation"
            type="t:VotingInformationType"
            minOccurs="0"
           />
          <xs:element name="ReminderMessageData"
            type="t:ReminderMessageDataType"
            minOccurs="0"
           />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

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

Element name	Type	Description
Sender	**t:SingleRecipientType** ([MS-OXWSCDATA] section 2.2.4.69)	Specifies the sender of a message. This element is optional. This is a read/write element.
ToRecipients	**t:ArrayOfRecipientsType** ([MS-OXWSCDATA] section 2.2.4.11)	Specifies a collection of recipients of an email. This element is required for sending a message. This is a read/write element.
CcRecipients	**t:ArrayOfRecipientsType**	Specifies a collection of recipients that receive a carbon copy (Cc) of an email. This element is optional. This is a read/write element.
BccRecipients	**t:ArrayOfRecipientsType**	Specifies a collection of recipients that receive a blind carbon copy (Bcc) of an email. This element is optional. This is a

Element name	Type	Description
		read/write element.
IsReadReceiptRequested	**xs:boolean** ([XMLSCHEMA2] sec 3.2.2)	Specifies a Boolean value that indicates whether the sender of a message requests a read receipt. This element is optional. This is a read/write element. A text value of "true" indicates that a read receipt is requested from the recipient of the message.
IsDeliveryReceiptRequested	**xs:boolean**	Specifies a Boolean value that indicates whether the sender of the message has requested a delivery receipt. This is a read/write element. A text value of "true" indicates that a delivery receipt has been requested from the recipient of the message.
ConversationIndex	**xs:base64Binary** ([XMLSCHEMA2] sec 3.2.16)	Specifies the position of the message within a conversation. This element is optional. This element is read-only.
ConversationTopic	**xs:string** ([XMLSCHEMA2] sec 3.2.1)	Specifies the subject of the conversation. This element is optional. This element is read-only.
From	**t:SingleRecipientType**	Specifies the addressee from whom the message was sent. This element is optional. This is a read/write element.
InternetMessageId	**xs:string**	Specifies the Internet message identifier for the message. This element is optional. This element is read-only.
IsRead	**xs:boolean**	Specifies a Boolean value that indicates whether the message has been read. This is a read/write element. The text value of "true" indicates that the message has been read.
IsResponseRequested	**xs:boolean**	Specifies a Boolean value that indicates whether a response to an email has been requested. This element is optional. This is a read/write element. A text value of "true" indicates that a response has been requested.
References	**xs:string**	Specifies 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**	Specifies a collection of addresses to send replies to. This element is optional. This is a read/write element.
ReceivedBy	**t:SingleRecipientType**	Identifies the **delegate** in a **delegate access** scenario. This element is read-only.
ReceivedRepresenting	**t:SingleRecipientType**	Identifies the principal in a delegate access scenario. This element is read-only.

Element name	Type	Description
ApprovalRequestData	**t:ApprovalRequestDataType** (section 2.2.4.1)	Specifies the approval state of an approval request message.<8> This element is read-only.
VotingInformation	**t:VotingInformationType** (section 2.2.4.5)	Specifies voting information on messages that include voting buttons or voting response messages.<9> This element is read-only.
ReminderMessageData	**t:ReminderMessageDataType** (section 2.2.4.4)	Specifies the data in a reminder message.<10> This element is read-only.

2.2.4.4 t:ReminderMessageDataType Complex Type

The **ReminderMessageDataType** complex type specifies a reminder message.<11>

```
<xs:complexType name="ReminderMessageDataType">
  <xs:sequence>
    <xs:element name="ReminderText" type="xs:string" minOccurs="0"/>
    <xs:element name="Location" type="xs:string" minOccurs="0"/>
    <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="EndTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="AssociatedCalendarItemId" type="t:ItemIdType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

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

Element	Type	Description
ReminderText	**xs:string** ([XMLSCHEMA2])	the reminder message text.
Location	**xs:string**	The location of the reminder's associated event.
StartTime	**xs:dateTime** ([XMLSCHEMA2])	The start time of the reminder's associated event.
EndTime	**xs:dateTime**	The end time of the reminder's associated event.
AssociatedCalendarItemId	**t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.25)	The item id of the reminder's associated event.

2.2.4.5 t:VotingInformationType Complex Type

The **VotingInformationType** complex type specifies voting information.<12>

```
<xs:complexType name="VotingInformationType">
  <xs:sequence>

    <xs:element name="UserOptions" type="t:ArrayOfVotingOptionDataType" minOccurs="0"/>
    <xs:element name="VotingResponse" type="xs:string" minOccurs="0"/>
```

```
  </xs:sequence>
/xs:complexType>
```

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

Element	Type	Description
UserOptions	**t:ArrayOfVotingOptionDataType** (section 2.2.4.2)	The list of voting options.
VotingResponse	**xs:string** ([XMLSCHEMA2])	The voting response.

2.2.4.6 t:VotingOptionDataType Complex Type

The **VotingOptionDataType** complex type specifies the voting option on a message.<13>

```
<xs:complexType name="VotingOptionDataType">
  <xs:sequence>
    <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
   <xs:element name="SendPrompt" type="t:SendPromptType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

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

Element	Type	Description
DisplayName	**xs:string** ([XMLSCHEMA2])	The display name for the voting choice.
SendPrompt	**t:SendPromptType** (section 2.2.5.2)	Indicates the prompt behavior when the user selects this voting choice.

2.2.5 Simple Types

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

Simple type name	Description
MessageDispositionType (section 2.2.5.1)	Specifies how a message item is handled after it is created or updated.
SendPromptType (section 2.2.5.2)	Specifies the prompt behavior associated with a voting option.<14>

2.2.5.1 t:MessageDispositionType Simple Type

The **MessageDispositionType** simple type 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>
```

The following table lists the values that are defined by the **MessageDispositionType** simple type.

Value	Meaning
SaveOnly	When used in the **CreateItemType** complex type ([MS-OXWSCORE] section 3.1.4.2.3.2), the email message item is saved in the folder that is specified by the **TargetFolderIdType** complex type ([MS-OXWSFOLD] section 2.2.4.16). Messages can be sent later by using the **SendItem** operation (section 3.1.4.6) on an **ExchangeServiceBinding** object. In this case, an item identifier is returned.
SendOnly	When used in the **CreateItemType** complex type, the email message item is sent but no copy is saved. In this case, an item identifier is not returned.
SendAndSaveCopy	When used in the **CreateItemType** complex type, the email message item is sent and a copy is saved in the **TargetFolderIdType** complex type. In this case, an item identifier is not returned.

2.2.5.2 t:SendPromptType Simple Type

The **SendPromptType** simple type specifies the prompt behavior associated with a voting option.<15>

```
<xs:simpleType name="SendPromptType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Send"/>
    <xs:enumeration value="VotingOption"/>
  </xs:restriction>
</xs:simpleType>
```

The following table lists the possible values for the **SendPromptType** simple type.

Value	Meaning
None	No prompt behavior is specified.

Value	Meaning
Send	The response is sent immediately when the user chooses a voting option.
VotingOption	The user is prompted to confirm their choice before sending a response.

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 Email Message Types Items Web Service Protocol defines a single port type with seven operations. The operations enable client implementations to get, create, delete, update, move, copy, and send messages in a user's **mailbox**.

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 Rules

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

Operation name	Description
CopyItem	Copies an email message on the server.
CreateItem	Creates email messages on the server.
DeleteItem	Deletes an email message from the server.
GetItem	Gets email messages from the server.
MoveItem	Moves an email message on the server.
SendItem	Sends an email message to the server.
UpdateItem	Updates an email message on the server.

3.1.4.1 CopyItem

The **CopyItem** operation copies email messages on the server.

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

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

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

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

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

If the **CopyItem** **WSDL operation** request is successful, the server returns a **CopyItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.1.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **CopyItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **CopyItemResponseMessage** element is set to "NoError".

If the **CopyItem** WSDL operation request is not successful, it returns a **CopyItemResponse** element with the **ResponseClass** attribute of the **CopyItemResponseMessage** element set to "Error". The **ResponseCode** element of the **CopyItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **CopyItem** as described in [MS-OXWSCORE] section 3.1.4.1.

3.1.4.2 CreateItem

The **CreateItem** operation creates email messages.

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

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

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

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

```
      <soap:header message="tns:CreateItemSoapIn" part="TimeZoneContext" use="literal"/>
      <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="CreateItemResult" use="literal" />
      <soap:header message="tns:CreateItemSoapOut" part="ServerVersion" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

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

If the **CreateItem** **WSDL operation** request is successful, the server returns a **CreateItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.2.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **CreateItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **CreateItemResponseMessage** element is set to "NoError".

If the **CreateItem** WSDL operation is not successful, it returns a **CreateItemResponse** element with the **ResponseClass** attribute of the **CreateItemResponseMessage** element set to "Error". The **ResponseCode** element of the **CreateItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **CreateItem** as described in [MS-OXWSCORE] section 3.1.4.2.

3.1.4.3 DeleteItem

The **DeleteItem** operation deletes email messages from the server store.

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

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

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

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

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

If the **DeleteItem** **WSDL operation** request is successful, the server returns a **DeleteItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.3.2.2, with the

ResponseClass attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **DeleteItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified by [MS-OXWSCDATA] section 2.2.4.65, of the **DeleteItemResponseMessage** element is set to "NoError".

If the **DeleteItem** WSDL operation request is not successful, it returns a **DeleteItemResponse** element with the **ResponseClass** attribute of the **DeleteItemResponseMessage** element set to "Error". The **ResponseCode** element of the **DeleteItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **DeleteItem** as described in [MS-OXWSCORE] section 3.1.4.3.

3.1.4.4 GetItem

The **GetItem** operation enables the user to get email messages and to access information about email messages.

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

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

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

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

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

Element name	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** attribute, 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 name	Description
BaseShape	Identifies the basic configuration of properties to be returned in an item response. Set

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

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

If the **GetItem** **WSDL operation** request is successful, the server returns a **GetItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.4.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **GetItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **GetItemResponseMessage** element is set to "NoError".

If the **GetItem** WSDL operation request is not successful, it returns a **GetItemResponse** element with the **ResponseClass** attribute of the **GetItemResponseMessage** element set to "Error". The **ResponseCode** element of the **GetItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **GetItem** as described in [MS-OXWSCORE] section 3.1.4.2.

3.1.4.5 MoveItem

The **MoveItem** operation moves one or more email messages to a single destination folder.

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

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

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

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

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

If the **MoveItem** **WSDL operation** request is successful, the server returns a **MoveItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.7.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **MoveItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **MoveItemResponseMessage** element is set to "NoError".

If the **MoveItem** WSDL operation request is not successful, it returns a **MoveItemResponse** element with the **ResponseClass** attribute of the **MoveItemResponseMessage** element set to "Error". The **ResponseCode** element of the **MoveItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **MoveItem** as described in [MS-OXWSCORE] section 3.1.4.7.

3.1.4.6 SendItem

The **SendItem** operation sends email messages that are located in the server store.

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

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

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

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

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

If the **SendItem** **WSDL operation** request is successful, the server returns a **SendItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.8.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **SendItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65 of the **SendItemResponseMessage** element is set to "NoError".

If the **SendItem** WSDL operation request is not successful, it returns a **SendItemResponse** element with the **ResponseClass** attribute of the **SendItemResponseMessage** element set to "Error". The **ResponseCode** element of the **SendItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **SendItem** as described in [MS-OXWSCORE] section 3.1.4.8.

3.1.4.7 UpdateItem

The **UpdateItem** operation updates email message properties in the server store.

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

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

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

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

The **UpdateItem** operation modifies a message that has already been created and sent. The **UpdateItem** operation request can contain the **MessageDispositionType** simple type (section 2.2.5.1) and MUST contain the **ConflictResolutionType** simple type ([MS-OXWSCORE] section 3.1.4.9.4.1).

The protocol client sends an **UpdateItemSoapIn** request **WSDL message**, and the protocol server responds with an **UpdateItemSoapOut** response WSDL message.

If the **UpdateItem** **WSDL operation** request is successful, the server returns an **UpdateItemResponse** element, as specified in [MS-OXWSCORE] section 3.1.4.9.2.2, with the **ResponseClass** attribute, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **UpdateItemResponseMessage** element, as specified in [MS-OXWSCDATA] section 2.2.4.12, set to "Success". The **ResponseCode** element, as specified in [MS-OXWSCDATA] section 2.2.4.65, of the **UpdateItemResponseMessage** element is set to "NoError".

If the **UpdateItem** WSDL operation request is not successful, it returns an **UpdateItemResponse** element with the **ResponseClass** attribute of the **UpdateItemResponseMessage** element set to "Error". The **ResponseCode** element of the **UpdateItemResponseMessage** element is set to one of the common errors defined in [MS-OXWSCDATA] section 2.2.5.24.

For more information, see **UpdateItem** as described in [MS-OXWSCORE] section 3.1.4.9.

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 that creates a message. This example creates an email message with only the **Subject** and **Body** properties completed, and then it sends the email message to User1 and User2 at Contoso.com. The email 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 email 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 that gets a message. This example gets a message from the server store. The message is already identified with its **ItemId** Id and **ChangeKey** attributes.

The client constructs the request **XML** and sends it to the server. Note that 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. Note that 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 that updates a message. This example updates the subject of an existing message. The message is already identified with its **ItemId** Id and **ChangeKey** attributes.

The client constructs the request **XML** and sends it to the server. Note that 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: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. Note that 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: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 that deletes a message. This example deletes the identified message from the server store.

The client constructs the request **XML** and sends it to the server. Note that 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 that moves a message to a specific folder. This example moves an identified message to the **Junk Email folder**.

The client constructs the request **XML** and sends it to the server. Note that the **ItemId** Id attribute has been shortened to preserve readability.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010" />
  </soap:Header>
  <soap:Body>
    <m: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. Note that 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: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 that copies a message to another folder. This example copies an identified message to the **Junk Email folder**.

The client constructs the request **XML** and sends it to the server. Note that 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: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. Note that 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: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 that sends a message to the server. This example sends an identified message.

The client constructs the request **XML** and sends it to the server. Note that 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:SendItem SaveItemToFolder="false">
        <m:ItemIds>
          <t:ItemId Id="AAMkAGIw" ChangeKey="CQAAABYAAA" />
        </m:ItemIds>
      </m:SendItem>
    </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: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>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

File name	Description	Section
MS-OXWSMSG.wsdl	Contains the **WSDL** for the implementation of this protocol.	6
MS-OXWSMSG-types.xsd	Contains the **XML schema** type definitions used in this protocol.	7
MS-OXWSCORE-messages.xsd	Contains XML schema message definitions used in this protocol.	[MS-OXWSCORE] section 7.1

These files have to be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSMSG-types.xsd schema or the MS-OXWSMSG-messages.xsd schema have to be placed in the common folder with these files.

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="Exchange2016"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
      <xs:include schemaLocation="MS-OXWSCORE-messages.xsd"/>
      <!-- Add global elements and types from messages.xsd -->
    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
      <!-- Add global elements and types from types.xsd -->
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="CopyItemSoapIn">
    <wsdl:part name="request" element="tns:CopyItem"/>
    <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="CopyItemSoapOut">
    <wsdl:part name="CopyItemResult" element="tns:CopyItemResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:message name="CreateItemSoapIn">
    <wsdl:part name="request" element="tns:CreateItem"/>
    <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="CreateItemSoapOut">
  <wsdl:part name="CreateItemResult" element="tns:CreateItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="DeleteItemSoapIn">
  <wsdl:part name="request" element="tns:DeleteItem"/>
  <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="DeleteItemSoapOut">
  <wsdl:part name="DeleteItemResult" element="tns:DeleteItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="GetItemSoapIn">
  <wsdl:part name="request" element="tns:GetItem"/>
  <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="GetItemSoapOut">
  <wsdl:part name="GetItemResult" element="tns:GetItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="MoveItemSoapIn">
  <wsdl:part name="request" element="tns:MoveItem"/>
  <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="MoveItemSoapOut">
  <wsdl:part name="MoveItemResult" element="tns:MoveItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="SendItemSoapIn">
  <wsdl:part name="request" element="tns:SendItem"/>
  <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="SendItemSoapOut">
  <wsdl:part name="SendItemResult" element="tns:SendItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="UpdateItemSoapIn">
  <wsdl:part name="request" element="tns:UpdateItem"/>
  <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="UpdateItemSoapOut">
  <wsdl:part name="UpdateItemResult" element="tns:UpdateItemResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:portType name="ExchangeServicePortType">
  <wsdl:operation name="CopyItem">
    <wsdl:input message="tns:CopyItemSoapIn"/>
    <wsdl:output message="tns:CopyItemSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="CreateItem">
    <wsdl:input message="tns:CreateItemSoapIn"/>
    <wsdl:output message="tns:CreateItemSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteItem">
    <wsdl:input message="tns:DeleteItemSoapIn"/>
```

```
      <wsdl:output message="tns:DeleteItemSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetItem">
      <wsdl:input message="tns:GetItemSoapIn"/>
      <wsdl:output message="tns:GetItemSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="MoveItem">
      <wsdl:input message="tns:MoveItemSoapIn"/>
      <wsdl:output message="tns:MoveItemSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="SendItem">
      <wsdl:input message="tns:SendItemSoapIn"/>
      <wsdl:output message="tns:SendItemSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="UpdateItem">
      <wsdl:input message="tns:UpdateItemSoapIn"/>
      <wsdl:output message="tns:UpdateItemSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:documentation>
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/"/>
    </wsdl:documentation>
    <wsdl:operation name="CopyItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CopyItem"/>
      <wsdl:input>
        <soap:header message="tns:CopyItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:CopyItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:CopyItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="CopyItemResult" use="literal"/>
        <soap:header message="tns:CopyItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateItem"/>
      <wsdl:input>
        <soap:header message="tns:CreateItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:CreateItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:CreateItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:header message="tns:CreateItemSoapIn" part="TimeZoneContext" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="CreateItemResult" use="literal"/>
        <soap:header message="tns:CreateItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteItem"/>
      <wsdl:input>
        <soap:header message="tns:DeleteItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:DeleteItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:DeleteItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="DeleteItemResult" use="literal"/>
        <soap:header message="tns:DeleteItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetItem">
```

```
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetItem"/>
      <wsdl:input>
        <soap:header message="tns:GetItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:GetItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:GetItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:header message="tns:GetItemSoapIn" part="TimeZoneContext" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="GetItemResult" use="literal"/>
        <soap:header message="tns:GetItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="MoveItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/MoveItem"/>
      <wsdl:input>
        <soap:header message="tns:MoveItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:MoveItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:MoveItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="MoveItemResult" use="literal"/>
        <soap:header message="tns:MoveItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="UpdateItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateItem"/>
      <wsdl:input>
        <soap:header message="tns:UpdateItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:UpdateItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:UpdateItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:header message="tns:UpdateItemSoapIn" part="TimeZoneContext" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="UpdateItemResult" use="literal"/>
        <soap:header message="tns:UpdateItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SendItem">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/SendItem"/>
      <wsdl:input>
        <soap:header message="tns:SendItemSoapIn" part="Impersonation" use="literal"/>
        <soap:header message="tns:SendItemSoapIn" part="MailboxCulture" use="literal"/>
        <soap:header message="tns:SendItemSoapIn" part="RequestVersion" use="literal"/>
        <soap:body parts="request" use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body parts="SendItemResult" use="literal"/>
        <soap:header message="tns:SendItemSoapOut" part="ServerVersion" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

7 Appendix B: Full XML Schema

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

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

File name	Defining specification
MS-OXWSCORE-types.xsd	[MS-OXWSCORE] section 7.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
        elementFormDefault="qualified" version="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCORE-types.xsd"/>
  <xs:complexType name="ArrayOfVotingOptionDataType">
    <xs:sequence>
      <xs:element name="VotingOptionData" type="t:VotingOptionDataType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ApprovalRequestDataType">
    <xs:sequence>
      <xs:element name="IsUndecidedApprovalRequest" type="xs:boolean" minOccurs="0"/>
      <xs:element name="ApprovalDecision" type="xs:int" minOccurs="0"/>
      <xs:element name="ApprovalDecisionMaker" type="xs:string" minOccurs="0"/>
      <xs:element name="ApprovalDecisionTime" type="xs:dateTime" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <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:element name="ApprovalRequestData" type="t:ApprovalRequestDataType"
minOccurs="0"/>
        <xs:element name="VotingInformation" type="t:VotingInformationType" minOccurs="0"/>
        <xs:element name="ReminderMessageData" type="t:ReminderMessageDataType"
minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ReminderMessageDataType">
    <xs:sequence>
      <xs:element name="ReminderText" type="xs:string" minOccurs="0"/>
      <xs:element name="Location" type="xs:string" minOccurs="0"/>
```

```
        <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="EndTime" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="AssociatedCalendarItemId" type="t:ItemIdType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="VotingInformationType">
      <xs:sequence>
        <xs:element name="UserOptions" type="t:ArrayOfVotingOptionDataType" minOccurs="0"/>
        <xs:element name="VotingResponse" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="VotingOptionDataType">
      <xs:sequence>
        <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
        <xs:element name="SendPrompt" type="t:SendPromptType" minOccurs="0"/>
      </xs:sequence>
    </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:simpleType name="SendPromptType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="None"/>
        <xs:enumeration value="Send"/>
        <xs:enumeration value="VotingOption"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:schema>
```

8 Appendix C: Product Behavior

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

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 2.2.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ApprovalRequestDataType** complex type. This type was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1).

<2> Section 2.2.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ArrayOfVotingOptionDataType** complex type**.** This type was introduced in Exchange 2013 SP1.

<3> Section 2.2.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ReminderMessageDataType** complex type. This type was introduced in Exchange 2013 SP1.

<4> Section 2.2.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **VotingInformationType** complex type. This type was introduced in Exchange 2013 SP1.

<5> Section 2.2.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **VotingOptionDataType** complex type. This type was introduced in Exchange 2013 SP1.

<6> Section 2.2.4.1: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ApprovalRequestDataType** complex type. This type was introduced in Exchange 2013 SP1.

<7> Section 2.2.4.2: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ArrayOfVotingOptionDataType** complex type**.** This type was introduced in Exchange 2013 SP1.

<8> Section 2.2.4.3: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ApprovalRequestData** element. This element was introduced in Exchange 2013 SP1.

<9> Section 2.2.4.3: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **VotingInformation** element. This element was introduced in Exchange 2013 SP1.

<10> Section 2.2.4.3: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ReminderMessageData** element. This element was introduced in Exchange 2013 SP1.

<11> Section 2.2.4.4: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **ReminderMessageDataType** complex type. This type was introduced in Exchange 2013 SP1.

<12> Section 2.2.4.5: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **VotingInformationType** complex type. This type was introduced in Exchange 2013 SP1.

<13> Section 2.2.4.6: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **VotingOptionDataType** complex type. This type was introduced in Exchange 2013 SP1.

<14> Section 2.2.5: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **SendPromptType** simple type. This type was introduced in Exchange 2013 SP1.

<15> Section 2.2.5.2: Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **SendPromptType** simple type. This type was introduced in Exchange 2013 SP1.

9 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

10 Index