

# [MS-OXWSNTIF]: Notifications Web Service Protocol

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

## Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	1.1.0	None	Version 1.1.0 release
5/5/2010	2.0.0	Major	Updated and revised the technical content.
8/4/2010	2.1	Minor	Clarified the meaning of the technical content.
11/3/2010	3.0	Major	Significantly changed the technical content.
3/18/2011	3.1	Minor	Clarified the meaning of the technical content.
8/5/2011	4.0	Major	Significantly changed the technical content.
10/7/2011	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	5.1	Minor	Clarified the meaning of the technical content.
10/8/2012	6.0	Major	Significantly changed the technical content.
2/11/2013	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	6.1	Minor	Clarified the meaning of the technical content.
2/10/2014	6.1	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	7.0	Major	Significantly changed the technical content.
7/31/2014	7.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	7.1	Minor	Clarified the meaning of the technical content.
5/26/2015	8.0	Major	Significantly changed the technical content.
9/14/2015	8.0	No Change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Glossary .....	5
1.2	References .....	6
1.2.1	Normative References .....	6
1.2.2	Informative References .....	7
1.3	Overview .....	7
1.4	Relationship to Other Protocols .....	7
1.5	Prerequisites/Preconditions .....	8
1.6	Applicability Statement .....	8
1.7	Versioning and Capability Negotiation .....	8
1.8	Vendor-Extensible Fields .....	8
1.9	Standards Assignments.....	8
<b>2</b>	<b>Messages.....</b>	<b>9</b>
2.1	Transport .....	9
2.2	Common Message Syntax .....	9
2.2.1	Namespaces .....	9
2.2.2	Messages.....	9
2.2.3	Elements .....	9
2.2.4	Complex Types.....	10
2.2.4.1	m:GetEventsResponseMessageType Complex Type .....	10
2.2.4.2	m:GetStreamingEventsResponseMessageType Complex Type .....	11
2.2.4.3	m:SubscribeResponseMessageType Complex Type .....	11
2.2.4.4	t:BaseNotificationEventType Complex Type .....	12
2.2.4.5	t:BaseObjectChangedEventType Complex Type .....	12
2.2.4.6	t:ModifiedEventType Complex Type .....	13
2.2.4.7	t:MovedCopiedEventType Complex Type .....	14
2.2.4.8	t:NotificationType Complex Type.....	15
2.2.5	Simple Types .....	16
2.2.5.1	t:WatermarkType Simple Type .....	16
2.2.5.2	t:SubscriptionIdType Simple Type .....	16
2.2.6	Attributes .....	17
2.2.7	Groups .....	17
2.2.8	Attribute Groups.....	17
<b>3</b>	<b>Protocol Details.....</b>	<b>18</b>
3.1	ExchangeServicePortType Server Details.....	18
3.1.1	Abstract Data Model.....	18
3.1.2	Timers .....	18
3.1.3	Initialization.....	18
3.1.4	Message Processing Events and Sequencing Rules .....	18
3.1.4.1	GetEvents .....	18
3.1.4.1.1	Messages .....	19
3.1.4.1.1.1	tns:GetEventsSoapIn Message.....	19
3.1.4.1.1.2	tns:GetEventsSoapOut Message .....	20
3.1.4.1.2	Elements .....	20
3.1.4.1.2.1	GetEvents Element.....	20
3.1.4.1.2.2	GetEventsResponse Element .....	20
3.1.4.1.3	Complex Types .....	21
3.1.4.1.3.1	m:GetEventsResponseType Complex Type .....	21
3.1.4.1.3.2	m:GetEventsType Complex Type .....	21
3.1.4.2	GetStreamingEvents.....	22
3.1.4.2.1	Messages .....	22
3.1.4.2.1.1	tns:GetStreamingEventsSoapIn .....	22
3.1.4.2.1.2	tns:GetStreamingEventsSoapOut .....	23

3.1.4.2.2	Elements .....	24
3.1.4.2.2.1	GetStreamingEvents Element .....	24
3.1.4.2.2.2	GetStreamingEventsResponse Element .....	24
3.1.4.2.3	Complex Types .....	24
3.1.4.2.3.1	m:GetStreamingEventsType Complex Type .....	24
3.1.4.2.3.2	m:GetStreamingEventsResponseType Complex Type .....	25
3.1.4.2.3.3	t:StreamingSubscriptionRequestType Complex Type .....	25
3.1.4.2.3.4	t:NonEmptyArrayOfSubscriptionIdsType Complex Type .....	26
3.1.4.2.3.5	t:NonEmptyArrayOfNotificationsType Complex Type .....	26
3.1.4.2.4	Simple Types .....	27
3.1.4.2.4.1	t:StreamingSubscriptionConnectionTimeoutType Simple Type .....	27
3.1.4.2.4.2	t:ConnectionStatusType Simple Type .....	27
3.1.4.3	Subscribe .....	28
3.1.4.3.1	Messages .....	28
3.1.4.3.1.1	tns:SubscribeSoapIn Message .....	29
3.1.4.3.1.2	tns:SubscribeSoapOut Message .....	29
3.1.4.3.2	Elements .....	29
3.1.4.3.2.1	Subscribe Element .....	30
3.1.4.3.2.2	SubscribeResponse Element .....	30
3.1.4.3.3	Complex Types .....	30
3.1.4.3.3.1	m:SubscribeResponseType Complex Type .....	30
3.1.4.3.3.2	m:SubscribeType Complex Type .....	31
3.1.4.3.3.3	t:BaseSubscriptionRequestType Complex Type .....	31
3.1.4.3.3.4	t:NonEmptyArrayOfNotificationEventTypesType Complex Type .....	32
3.1.4.3.3.5	t:PullSubscriptionRequestType Complex Type .....	33
3.1.4.3.3.6	t:PushSubscriptionRequestType Complex Type .....	33
3.1.4.3.4	Simple Types .....	34
3.1.4.3.4.1	t:NotificationEventTypeType Simple Type .....	34
3.1.4.3.4.2	t:SubscriptionTimeoutType Simple Type .....	35
3.1.4.3.4.3	t:SubscriptionStatusFrequencyType Simple Type .....	35
3.1.4.4	Unsubscribe .....	36
3.1.4.4.1	Messages .....	36
3.1.4.4.1.1	tns:UnsubscribeSoapIn Message .....	36
3.1.4.4.1.2	tns:UnsubscribeSoapOut Message .....	37
3.1.4.4.2	Elements .....	37
3.1.4.4.2.1	Unsubscribe Element .....	37
3.1.4.4.2.2	UnsubscribeResponse Element .....	38
3.1.4.4.3	Complex Types .....	38
3.1.4.4.3.1	m:UnsubscribeResponseType Complex Type .....	38
3.1.4.4.3.2	m:UnsubscribeType Complex Type .....	38
3.1.5	Timer Events .....	39
3.1.6	Other Local Events .....	39
<b>4</b>	<b>Protocol Examples .....</b>	<b>40</b>
<b>5</b>	<b>Security .....</b>	<b>41</b>
5.1	Security Considerations for Implementers .....	41
5.2	Index of Security Parameters .....	41
<b>6</b>	<b>Appendix A: Full WSDL .....</b>	<b>42</b>
<b>7</b>	<b>Appendix B: Full XML Schema .....</b>	<b>45</b>
7.1	Messages Schema .....	45
7.2	Types Schema .....	47
<b>8</b>	<b>Appendix C: Product Behavior .....</b>	<b>50</b>
<b>9</b>	<b>Change Tracking .....</b>	<b>52</b>
<b>10</b>	<b>Index .....</b>	<b>53</b>



# 1 Introduction

The Notifications Web Service Protocol enables clients to receive pull notifications from the server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [\[RFC2119\]](#). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are specific to this document:

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

**free/busy status:** A property of an appointment that indicates how an appointment on the calendar of an attendee or resource affects their availability.

**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 [\[RFC5246\]](#).

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

**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 [\[SOAP1.2-1/2003\]](#).

**SOAP action:** The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

**SOAP body:** A container for the payload data being delivered by a SOAP message to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

**SOAP header:** A mechanism for implementing extensions to a SOAP message in a decentralized manner without prior agreement between the communicating parties. See [\[SOAP1.2-1/2007\]](#) section 5.2 for more information.

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

**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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[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, <http://www.rfc-editor.org/rfc/rfc2119.txt>

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

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", BCP 47, RFC 3066, January 2001, <http://www.ietf.org/rfc/rfc3066.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://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, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

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

### 1.2.2 Informative References

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol](#)".

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

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

[MS-OXWSPSNTIF] Microsoft Corporation, "[Push Notifications Web Service Protocol](#)".

### 1.3 Overview

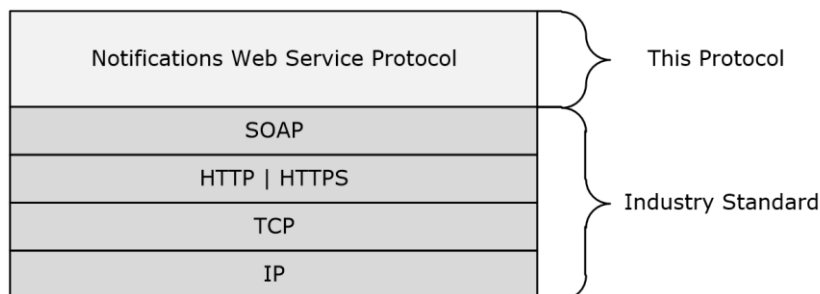
This protocol enables clients to subscribe to receive pull notifications from the server, or to receive streaming notifications from the server. For pull notifications, clients subscribe to updates and then receive these updates from the server in response to specific requests. For streaming notifications, the server automatically pushes the updates to clients through an open connection.

### 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 [\[MS-OXDCLI\]](#), to identify the target **endpoint** to use for each operation.

This protocol uses the SOAP Protocol, as described in [\[SOAP1.1\]](#), to specify the structure information exchanged between the client and the 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 layering diagram.



**Figure 1: This protocol in relation to other protocols**

The information that is returned by this protocol is used when requests are made by using the Push Notifications Web Service Protocol, as described in [\[MS-OXWSPSNTIF\]](#).

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 described in [\[MS-OXWSADISC\]](#), or the Autodiscover Publishing and Lookup Protocol, as described in [\[MS-OXDSCLI\]](#), 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.

## 1.6 Applicability Statement

The protocol specified in this document is applicable to environments that incorporate notifications.

## 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 specified in section [2.1](#).
- **Protocol Versions:** This protocol has only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.11, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).
- **Capability Negotiation:** 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

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

This protocol relies on the Web server that hosts the application to perform authentication. This protocol **MUST** support SOAP over HTTP, as specified in [\[RFC2616\]](#). The protocol **SHOULD** use secure communications by means of 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 defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and WSDL as defined in [\[WSDL\]](#).

#### 2.2.1 Namespaces

This specification defines and references various **XML namespace** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

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

#### 2.2.2 Messages

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

#### 2.2.3 Elements

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

## 2.2.4 Complex Types

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

Complex type name	Description
<b>BaseNotificationEventType</b>	Specifies the base type for event notifications.
<b>BaseObjectChangedEventType</b>	Specifies identification and time stamp details for create, delete, and new mail events.
<b>GetEventsResponseMessageType</b>	Specifies the status and result of a single <b>GetEvents</b> request.
<b>GetStreamingEventsResponseMessageType</b>	Specifies the status and result of a single <b>GetStreamingEvents</b> operation request.
<b>ModifiedEventType</b>	Specifies an event in which an item or folder is modified.
<b>MovedCopiedEventType</b>	Specifies a moved or copied event notification.
<b>NotificationType</b>	Specifies information about the subscription and the events that have occurred since the last notification.
<b>SubscribeResponseMessageType</b>	Specifies the status and result of a single <b>Subscribe</b> request.

### 2.2.4.1 m:GetEventsResponseMessageType Complex Type

The **GetEventsResponseMessageType** complex type specifies the status and result of a single **GetEvents** request. The **GetEventsResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.67.

```
<xs:complexType name="GetEventsResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Notification"
          type="t:NotificationType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **GetEventsResponseMessageType** complex type.

Element name	Type	Description
<b>Notification</b>	<b>t:NotificationType</b> (section <a href="#">2.2.4.8</a> )	Specifies information about the subscription and the events that have occurred since the last notification.

### 2.2.4.2 m:GetStreamingEventsResponseMessageType Complex Type

The **GetStreamingEventsResponseMessageType** complex type specifies the status and result of a single **GetStreamingEvents** operation request. [<1>](#) The **GetStreamingEventsResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.67.

```
<xs:complexType name="GetStreamingEventsResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Notifications"
          type="t:NonEmptyArrayOfNotificationsType"
          minOccurs="0"
        />
        <xs:element name="ErrorSubscriptionIds"
          type="t:NonEmptyArrayOfSubscriptionIdsType"
          minOccurs="0"
        />
        <xs:element name="ConnectionStatus"
          type="t:ConnectionStatusType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **GetStreamingEventsResponseMessageType** complex type.

Element name	Type	Description
<b>Notifications</b>	<b>NonEmptyArrayOfNotificationsType</b> (section <a href="#">3.1.4.2.3.5</a> )	Specifies a list of information about the subscription and the events that have occurred since the last notification.
<b>ErrorSubscriptionIds</b>	<b>NonEmptyArrayOfSubscriptionIdsType</b> (section <a href="#">3.1.4.2.3.4</a> )	Specifies a list of subscription IDs that are invalid.
<b>ConnectionStatus</b>	<b>ConnectionStatusType</b> (section <a href="#">3.1.4.2.4.2</a> )	Specifies a text description of the status of a streaming subscription.

### 2.2.4.3 m:SubscribeResponseMessageType Complex Type

The **SubscribeResponseMessageType** complex type specifies the status and result of a single **Subscribe** operation request, as specified in section [3.1.4.3](#). The **SubscribeResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.67.

```
<xs:complexType name="SubscribeResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="SubscriptionId"

```

```

        type="t:SubscriptionIdType"
        minOccurs="0"
      />
      <xs:element name="Watermark"
        type="t:WatermarkType"
        minOccurs="0"
      />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table lists and describes the child elements of the **SubscribeResponseMessageType** complex type.

Element name	Type	Description
<b>SubscriptionId</b>	<b>t:SubscriptionIdType</b> (section <a href="#">2.2.5.2</a> )	Specifies the identifier for a subscription.
<b>Watermark</b>	<b>t:WatermarkType</b> (section <a href="#">2.2.5.1</a> )	Specifies an event bookmark in the <b>mailbox</b> event queue.

#### 2.2.4.4 t:BaseNotificationEventType Complex Type

The **BaseNotificationEventType** complex type specifies the base type for event notifications.

```

<xs:complexType name="BaseNotificationEventType">
  <xs:sequence>
    <xs:element name="Watermark"
      type="t:WatermarkType"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>

```

The following table lists and describes the child elements of the **BaseNotificationEventType** complex type.

Element name	Type	Description
<b>Watermark</b>	<b>t:WatermarkType</b> (section <a href="#">2.2.5.1</a> )	Specifies an event bookmark in the mailbox event queue. <2>

#### 2.2.4.5 t:BaseObjectChangedEventType Complex Type

The **BaseObjectChangedEventType** complex type specifies identification and time stamp details for create, delete, and new mail events. The **BaseObjectChangedEventType** complex type extends the **BaseNotificationEventType** complex type, as specified in section [2.2.4.4](#).

```

<xs:complexType name="BaseObjectChangedEventType">

```

```

<xs:complexContent>
  <xs:extension
    base="t:BaseNotificationEventType"
  >
    <xs:sequence>
      <xs:element name="TimeStamp"
        type="xs:dateTime"
      />
      <xs:choice>
        <xs:element name="FolderId"
          type="t:FolderIdType"
        />
        <xs:element name="ItemId"
          type="t:ItemIdType"
        />
      </xs:choice>
      <xs:element name="ParentFolderId"
        type="t:FolderIdType"
      />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table lists and describes the child elements of the **BaseObjectChangedEventType** complex type.

Element name	Type	Description
<b>TimeStamp</b>	<b>xs:dateTime</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the time stamp of a changed item or folder mailbox event.
<b>FolderId</b>	<b>t:FolderIdType</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.36)	Specifies a <b>FolderIdType</b> object that identifies the changed store item.
<b>ItemId</b>	<b>t:ItemIdType</b> ( <a href="#">[MS-OXWSCORE]</a> section 2.2.4.25)	Specifies an <b>ItemIdType</b> object that identifies the changed store item.
<b>ParentFolderId</b>	<b>t:FolderIdType</b>	Specifies the parent folder identifier of the changed item or folder.

### 2.2.4.6 t:ModifiedEventType Complex Type

The **ModifiedEventType** complex type specifies an event in which an item or folder is modified. The **ModifiedEventType** complex type extends the **BaseObjectChangedEventType** complex type, as specified in section [2.2.4.5](#).

```

<xs:complexType name="ModifiedEventType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseObjectChangedEventType"
    >
      <xs:sequence>
        <xs:element name="UnreadCount"
          type="xs:int"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

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

The following table lists and describes the child elements of the **ModifiedEventType** complex type.

Element name	Type	Description
<b>UnreadCount</b>	<b>xs:int</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the count of unread items in a particular folder. <a href="#">&lt;3&gt;</a>

#### 2.2.4.7 t:MovedCopiedEventType Complex Type

The **MovedCopiedEventType** complex type specifies a moved or copied event notification. The **MovedCopiedEventType** complex type extends the **BaseObjectChangedEventType** complex type, as specified in section [2.2.4.5](#).

```
<xs:complexType name="MovedCopiedEventType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseObjectChangedEventType"
    >
      <xs:sequence>
        <xs:choice>
          <xs:element name="OldFolderId"
            type="t:FolderIdType"
          />
          <xs:element name="OldItemId"
            type="t:ItemIdType"
          />
        </xs:choice>
        <xs:element name="OldParentFolderId"
          type="t:FolderIdType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **MovedCopiedEventType** complex type.

Element name	Type	Description
<b>OldFolderId</b>	<b>t:FolderIdType</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.4.36)	Specifies the old identifier for a copied or moved folder.
<b>OldItemId</b>	<b>t:ItemIdType</b> ( <a href="#">[MS-OXWSCORE]</a> section 2.2.4.25)	Specifies the old identifier for a copied or moved item.
<b>OldParentFolderId</b>	<b>t:FolderIdType</b>	Specifies the identifier of the original parent folder of a moved item or folder.

## 2.2.4.8 t:NotificationType Complex Type

The **NotificationType** complex type specifies information about the subscription and the events that have occurred since the last notification.

```
<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element name="SubscriptionId"
      type="t:SubscriptionIdType"
    />
    <xs:element name="PreviousWatermark"
      type="t:WatermarkType"
      minOccurs="0"
    />
    <xs:element name="MoreEvents"
      type="xs:boolean"
      minOccurs="0"
    />
    <xs:choice
      maxOccurs="unbounded"
    >
      <xs:element name="CopiedEvent"
        type="t:MovedCopiedEventType"
      />
      <xs:element name="CreatedEvent"
        type="t:BaseObjectChangedEventType"
      />
      <xs:element name="DeletedEvent"
        type="t:BaseObjectChangedEventType"
      />
      <xs:element name="ModifiedEvent"
        type="t:ModifiedEventType"
      />
      <xs:element name="MovedEvent"
        type="t:MovedCopiedEventType"
      />
      <xs:element name="NewMailEvent"
        type="t:BaseObjectChangedEventType"
      />
      <xs:element name="StatusEvent"
        type="t:BaseNotificationEventType"
      />
      <xs:element name="FreeBusyChangedEvent"
        type="t:BaseObjectChangedEventType"
      />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

The following table lists and describes the child elements of the **NotificationType** complex type.

Element name	Type	Description
<b>SubscriptionId</b>	<b>t:SubscriptionIdType</b> (section <a href="#">2.2.5.2</a> )	Specifies the identifier for a subscription.
<b>PreviousWatermark</b>	<b>t:WatermarkType</b> (section <a href="#">2.2.5.1</a> )	Specifies the watermark of the latest event that was successfully communicated to the client for the subscription. <a href="#">&lt;4&gt;</a>
<b>MoreEvents</b>	<b>xs:boolean</b> <a href="#">[XMLSCHEMA2]</a>	Specifies a value that indicates whether more events to be delivered to the client currently exist in the queue. <a href="#">&lt;5&gt;</a>



Element name	Type	Description
<b>CopiedEvent</b>	<b>t:MovedCopiedEventType</b> (section <a href="#">2.2.4.7</a> )	Specifies a copied event notification. <a href="#">&lt;6&gt;</a>
<b>CreatedEvent</b>	<b>t:BaseObjectChangedEventType</b> (section <a href="#">2.2.4.5</a> )	Specifies a created event notification.
<b>DeletedEvent</b>	<b>t:BaseObjectChangedEventType</b>	Specifies a deleted event notification.
<b>ModifiedEvent</b>	<b>t:ModifiedEventType</b> (section <a href="#">2.2.4.6</a> )	Specifies a modified event notification.
<b>MovedEvent</b>	<b>t:MovedCopiedEventType</b>	Specifies a moved event notification.
<b>NewMailEvent</b>	<b>t:BaseObjectChangedEventType</b>	Specifies a new e-mail event notification.
<b>StatusEvent</b>	<b>t:BaseNotificationEventType</b> (section <a href="#">2.2.4.4</a> )	Specifies a status event notification.
<b>FreeBusyChangedEvent</b>	<b>t:BaseObjectChangedEventType</b>	Specifies an event in which an item's <b>free/busy status</b> has changed. <a href="#">&lt;7&gt;</a>

## 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 defined with the operation.

Simple type name	Description
<b>WatermarkType</b>	Specifies an event bookmark in the mailbox event queue.
<b>SubscriptionIdType</b>	Specifies the identifier for a subscription.

### 2.2.5.1 t:WatermarkType Simple Type

The **WatermarkType** simple type specifies an event bookmark in the mailbox event queue.

```
<xs:simpleType name="WatermarkType">
  <xs:restriction
    base="t:NonEmptyStringType"
  />
</xs:simpleType>
```

### 2.2.5.2 t:SubscriptionIdType Simple Type

The **SubscriptionIdType** simple type specifies the identifier for a subscription.

```
<xs:simpleType name="SubscriptionIdType">
  <xs:restriction
    base="t:NonEmptyStringType"
  />
</xs:simpleType>
```

### **2.2.6 Attributes**

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

### **2.2.7 Groups**

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

### **2.2.8 Attribute Groups**

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

## 3 Protocol Details

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

### 3.1 ExchangeServicePortType Server Details

This protocol defines a single port type with four operations. These operations enable client implementations to get events as well as to subscribe to and unsubscribe from notifications.

#### 3.1.1 Abstract Data Model

None.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing Rules

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

Operation name	Description
<b>GetEvents</b>	Gets events for a pull notification subscription.
<b>GetStreamingEvents</b>	Gets events for a streaming notification subscription.
<b>Subscribe</b>	Creates a subscription to notifications of events on mailboxes.
<b>Unsubscribe</b>	Removes a subscription to notifications of events on mailboxes.

##### 3.1.4.1 GetEvents

The **GetEvents** operation gets events for a pull notification subscription.

The following is the WSDL port type specification for the **GetEvents** operation.

```
<wsdl:operation name="GetEvents" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetEventsSoapIn" />
  <wsdl:output message="tns:GetEventsSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification for the **GetEvents** operation.

```
<wsdl:operation name="GetEvents">
```

```

<soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents" />
<wsdl:input>
  <soap:header message="tns:GetEventsSoapIn" part="Impersonation" use="literal"/>
  <soap:header message="tns:GetEventsSoapIn" part="MailboxCulture" use="literal"/>
  <soap:header message="tns:GetEventsSoapIn" part="RequestVersion" use="literal"/>
  <soap:body parts="request" use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body parts="GetEventsResult" use="literal" />
  <soap:header message="tns:GetEventsSoapOut" part="ServerVersion" use="literal"/>
</wsdl:output>
</wsdl:operation>

```

### 3.1.4.1.1 Messages

The following table lists the WSDL message definitions are specific to the **GetEvents** operation.

Message name	Description
<b>GetEventsSoapIn</b>	Specifies the <b>GetEvents</b> operation WSDL message.
<b>GetEventsSoapOut</b>	Specifies the <b>GetEvents</b> operation response WSDL message.

#### 3.1.4.1.1.1 tns:GetEventsSoapIn Message

The **GetEventsSoapIn** WSDL message specifies the **GetEvents** operation request to get the events for a pull notification.

```

<wsdl:message name="GetEventsSoapIn">
  <wsdl:part name="request" element="tns:GetEvents" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>

```

The **GetEventsSoapIn** WSDL message is the input message for the **SOAP action** <http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents>.

The parts of the **GetEventsSoapIn** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>request</b>	<b>tns:GetEvents</b> (section <a href="#">3.1.4.1.2.1</a> )	Specifies the <b>SOAP body</b> of the request.
<b>Impersonation</b>	<b>t:ExchangeImpersonation</b> ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a <b>SOAP header</b> that identifies the user whom the client application is impersonating.
<b>MailboxCulture</b>	<b>t:MailboxCulture</b> ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in <a href="#">[RFC3066]</a> .
<b>RequestVersion</b>	<b>t:RequestServerVersion</b> ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the <b>GetEvents</b> operation (section <a href="#">3.1.4.1</a> ) request.

### 3.1.4.1.1.2 tns:GetEventsSoapOut Message

The **GetEventsSoapOut** WSDL message specifies the server response to the **GetEvents** operation request to get the events for a pull notification.

```
<wsdl:message name="GetEventsSoapOut">
  <wsdl:part name="GetEventsResult" element="tns:GetEventsResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **GetEventsSoapOut** WSDL message is the output for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents>.

The parts of the **GetEventsSoapOut** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>GetEventsResult</b>	<b>tns:GetEventsResponse</b> (section <a href="#">3.1.4.1.2.2</a> )	Specifies the SOAP body of the response message.
<b>ServerVersion</b>	<b>t:ServerVersionInfo</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.12)	Specifies a SOAP header that identifies the schema version for the <b>GetEventsSoapOut</b> message.

### 3.1.4.1.2 Elements

The following table lists the XML schema element definitions that are specific to the **GetEvents** operation.

Element name	Description
<b>GetEvents</b>	Specifies a request to the <b>GetEvents</b> operation that is used by pull clients to request notifications from the server.
<b>GetEventsResponse</b>	Specifies a response to the <b>GetEvents</b> operation request.

#### 3.1.4.1.2.1 GetEvents Element

The **GetEvents** element specifies the operation that is used by pull clients to request notifications from the server.

```
<xs:element name="GetEvents"
  type="m:GetEventsType"
/>
```

#### 3.1.4.1.2.2 GetEventsResponse Element

The **GetEventsResponse** element specifies a response to a **GetEvents** operation request.

```
<xs:element name="GetEventsResponse"
  type="m:GetEventsResponseType"
/>
```

### 3.1.4.1.3 Complex Types

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

Complex type name	Description
<b>GetEventsResponseType</b>	Specifies the response from the <b>GetEvents</b> operation.
<b>GetEventsType</b>	Specifies a request to get events for a pull notification subscription.

#### 3.1.4.1.3.1 m:GetEventsResponseType Complex Type

The **GetEventsResponseType** complex type specifies a response to a **GetEvents** operation request. The **GetEventsResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

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

#### 3.1.4.1.3.2 m:GetEventsType Complex Type

The **GetEventsType** complex type specifies a request to get events for a pull notification subscription. The **GetEventsType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="GetEventsType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
    <xs:sequence>
      <xs:element name="SubscriptionId"
        type="t:SubscriptionIdType"
      />
      <xs:element name="Watermark"
        type="t:WatermarkType"
      />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **GetEventsType** complex type.

Element name	Type	Description
<b>SubscriptionId</b>	<b>t:SubscriptionIdType</b> (section <a href="#">2.2.5.2</a> )	Specifies the identifier of a subscription.
<b>Watermark</b>	<b>t:WatermarkType</b> (section <a href="#">2.2.5.1</a> )	Specifies a string that represents an event bookmark in the mailbox event queue.

### 3.1.4.2 GetStreamingEvents

The **GetStreamingEvents** operation gets events for a streaming notification subscription. [<8>](#)

The following is the WSDL port type specification for the **GetStreamingEvents** operation.

```
<wsdl:operation name="GetStreamingEvents">
  <wsdl:input message="tns:GetStreamingEventsSoapIn" />
  <wsdl:output message="tns:GetStreamingEventsSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification for the **GetStreamingEvents** operation.

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

#### 3.1.4.2.1 Messages

The following table lists the WSDL message definitions that are specific to the **GetStreamingEvents** operation.

Message name	Description
<b>GetStreamingEventsSoapIn</b>	Specifies the <b>GetStreamingEvents</b> operation WSDL message.
<b>GetStreamingEventsSoapOut</b>	Specifies the <b>GetStreamingEvents</b> operation response WSDL message.

##### 3.1.4.2.1.1 tns:GetStreamingEventsSoapIn

The **GetStreamingEventsSoapIn** WSDL message specifies the **GetStreamingEvents** operation request to get streaming event notifications.

```

<wsdl:message name="GetStreamingEventsSoapIn">
  <wsdl:part name="request" element="tns:GetStreamingEvents" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>

```

The **GetStreamingEventsSoapIn** WSDL message is the input message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetStreamingEvents>.

The parts of the **GetStreamingEventsSoapIn** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>request</b>	<b>tns:GetStreamingEvents</b> (section <a href="#">3.1.4.2.2.1</a> )	Specifies the SOAP body of the request.
<b>Impersonation</b>	<b>t:ExchangeImpersonation</b> ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
<b>MailboxCulture</b>	<b>t:MailboxCulture</b> ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in <a href="#">RFC3066</a> .
<b>RequestVersion</b>	<b>t:RequestServerVersion</b> ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the <b>GetStreamingEvents</b> operation request.

### 3.1.4.2.1.2 tns:GetStreamingEventsSoapOut

The **GetStreamingEventsSoapOut** WSDL message specifies the server response to the **GetStreamingEvents** operation request to get streaming event notifications.

```

<wsdl:message name="GetStreamingEventsSoapOut">
  <wsdl:part name="GetStreamingEventsResult" element="tns:GetStreamingEventsResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>

```

The **GetStreamingEventsSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetStreamingEvents>.

The parts of the **GetStreamingEventsSoapOut** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>GetStreamingEventsResult</b>	<b>GetStreamingEventsResponse</b> (section <a href="#">3.1.4.2.2.2</a> )	Specifies the SOAP body of the response message.
<b>ServerVersion</b>	<b>t:ServerVersionInfo</b> ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the schema version for the <b>GetStreamingEvents</b> operation (section <a href="#">3.1.4.2</a> ) message.



### 3.1.4.2.2 Elements

The following table lists the XML schema element definitions that are specific to the **GetStreamingEvents** operation.

Message name	Description
<b>GetStreamingEvents</b>	Specifies the operation that is used by clients to request streaming notifications from the server.
<b>GetStreamingEventsResponse</b>	Specifies a response to a <b>GetStreamingEvents</b> operation request.

#### 3.1.4.2.2.1 GetStreamingEvents Element

The **GetStreamingEvents** element specifies the operation that is used by clients to request streaming notifications from the server.

```
<xs:element name="GetStreamingEvents"  
  type="m:GetStreamingEventsType"  
/>
```

#### 3.1.4.2.2.2 GetStreamingEventsResponse Element

The **GetStreamingEventsResponse** element specifies a response to a **GetStreamingEvents** operation request.

```
<xs:element name="GetStreamingEventsResponse"  
  type="m:GetStreamingEventsResponseType"  
/>
```

#### 3.1.4.2.3 Complex Types

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

Message name	Description
<b>GetStreamingEventsType</b>	Specifies a request to get streaming notifications from the server.
<b>GetStreamingEventsResponseType</b>	Specifies a response from the <b>GetStreamingEvents</b> operation.
<b>StreamingSubscriptionRequestType</b>	Specifies a subscription to streaming event notifications.
<b>NonEmptyArrayOfSubscriptionIdsType</b>	Specifies an array of invalid subscription IDs.
<b>NonEmptyArrayOfNotificationsType</b>	Specifies an array of information about the subscription and the events that have occurred since the last notification.

##### 3.1.4.2.3.1 m:GetStreamingEventsType Complex Type

The **GetStreamingEventsType** complex type specifies a request to get streaming notifications from the server. The **GetStreamingEventsType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType>
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="SubscriptionIds"
          type="t:NonEmptyArrayOfSubscriptionIdsType"
        />
        <xs:element name="ConnectionTimeout"
          type="t:StreamingSubscriptionConnectionTimeoutType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **GetStreamingEventsType** complex type.

Element name	Type	Description
<b>SubscriptionIds</b>	<b>t:NonEmptyArrayOfSubscriptionIdsType</b> (section <a href="#">3.1.4.2.3.4</a> )	Specifies the identifier for a subscription that is queried for events.
<b>ConnectionTimeout</b>	<b>t:StreamingSubscriptionConnectionTimeoutType</b> (section <a href="#">3.1.4.2.4.1</a> )	Specifies the number of minutes to keep a connection open.

### 3.1.4.2.3.2 m:GetStreamingEventsResponseType Complex Type

The **GetStreamingEventsResponseType** complex type represents a response to a **GetStreamingEvents** operation request. The **GetStreamingEventsResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

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

### 3.1.4.2.3.3 t:StreamingSubscriptionRequestType Complex Type

The **StreamingSubscriptionRequestType** complex type specifies a subscription to streaming event notifications.

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

```

<xs:element name="FolderIds"
  type="t:NonEmptyArrayOfBaseFolderIdsType"
  minOccurs="0"
/>
<xs:element name="EventTypes"
  type="t:NonEmptyArrayOfNotificationEventTypesType"
/>
</xs:sequence>
</xs:complexType>

```

The following table lists and describes the child elements of the **StreamingSubscriptionRequestType** complex type.

Element name	Type	Description
<b>FolderIds</b>	<b>t:NonEmptyArrayOfBaseFolderIdsType</b> ( <a href="#">[MS-OXWSFOLD]</a> section 3.1.4.6.3.3)	Specifies an array of folder identifiers that are used to identify folders to be monitored for event notifications.
<b>EventTypes</b>	<b>t:NonEmptyArrayOfNotificationEventTypesType</b> (section <a href="#">3.1.4.3.3.4</a> )	Specifies a collection of event notifications that are used to create a subscription.

#### 3.1.4.2.3.4 t:NonEmptyArrayOfSubscriptionIdsType Complex Type

The **NonEmptyArrayOfSubscriptionIdsType** complex type specifies an array of subscription IDs.

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="SubscriptionId"
      type="t:SubscriptionIdType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>

```

The following table lists and describes the child elements of the **NonEmptyArrayOfSubscriptionIdsType** complex type.

Element name	Type	Description
<b>SubscriptionId</b>	<b>t:SubscriptionIdType</b> (section <a href="#">2.2.5.2</a> )	Specifies the identifier for a subscription.

#### 3.1.4.2.3.5 t:NonEmptyArrayOfNotificationsType Complex Type

The **NonEmptyArrayOfNotificationsType** complex type specifies an array of information about the subscription and the events that have occurred since the last notification.

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Notification"
      type="t:NotificationType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>

```

```

        minOccurs="0"
      />
    </xs:sequence>
  </xs:complexType>

```

The following table lists and describes the child elements of the **NonEmptyArrayOfNotificationsType** complex type.

Element name	Type	Description
<b>Notification</b>	<b>t:NotificationType</b> (section <a href="#">2.2.4.8</a> )	Specifies information about the subscription and the events that have occurred since the last notification.

### 3.1.4.2.4 Simple Types

The following table lists the XML schema simple type definitions that are specific to the **GetStreamingEvents** operation.

Simple type name	Description
<b>StreamingSubscriptionConnectionTimeoutType</b>	Specifies the number of minutes to keep a connection open.
<b>ConnectionStatusType</b>	Specifies a text description of the status of a streaming subscription.

#### 3.1.4.2.4.1 t:StreamingSubscriptionConnectionTimeoutType Simple Type

The **StreamingSubscriptionConnectionTimeoutType** simple type specifies the number of minutes to keep a connection open.

```

<xs:simpleType name="StreamingSubscriptionConnectionTimeoutType">
  <xs:restriction
    base="xs:int"
  >
    <xs:minInclusive
      value="1"
    />
    <xs:maxInclusive
      value="30"
    />
  </xs:restriction>
</xs:simpleType>

```

#### 3.1.4.2.4.2 t:ConnectionStatusType Simple Type

The **ConnectionStatusType** simple type specifies a text description of the status of a streaming subscription.

```

<xs:simpleType name="ConnectionStatusType">
  <xs:restriction

```

```

    base="xs:string"
  >
  <xs:enumeration
    value="OK"
  />
  <xs:enumeration
    value="Closed"
  />
</xs:restriction>
</xs:simpleType>

```

The following table lists and describes the values that are defined by the **ConnectionStatusType** simple type.

Value name	Meaning
<b>OK</b>	The connection is good.
<b>Closed</b>	The connection is closed.

### 3.1.4.3 Subscribe

The **Subscribe** operation creates a subscription to notifications of events on mailboxes on the server.

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

```

<wsdl:operation name="Subscribe" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:SubscribeSoapIn" />
  <wsdl:output message="tns:SubscribeSoapOut" />
</wsdl:operation>

```

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

```

<wsdl:operation name="Subscribe">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/Subscribe" />
  <wsdl:input>
    <soap:header message="tns:SubscribeSoapIn" part="Impersonation" use="literal"/>
    <soap:header message="tns:SubscribeSoapIn" part="MailboxCulture" use="literal"/>
    <soap:header message="tns:SubscribeSoapIn" part="RequestVersion" use="literal"/>
    <soap:body parts="request" use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="SubscribeResult" use="literal" />
    <soap:header message="tns:SubscribeSoapOut" part="ServerVersion" use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

#### 3.1.4.3.1 Messages

The following table lists the WSDL message definitions are specific to the **Subscribe** operation.

Message name	Description
<b>SubscribeSoapIn</b>	Specifies the <b>Subscribe</b> operation WSDL message.

Message name	Description
<b>SubscribeSoapOut</b>	Specifies the <b>Subscribe</b> operation response WSDL message.

### 3.1.4.3.1.1 tns:SubscribeSoapIn Message

The parts of the **SubscribeSoapIn** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>request</b>	<b>tns:Subscribe</b> (section <a href="#">3.1.4.3.2.1</a> )	Specifies the SOAP body of the request.
<b>Impersonation</b>	<b>t:ExchangeImpersonation</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
<b>MailboxCulture</b>	<b>t:MailboxCulture</b> ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in <a href="#">[RFC3066]</a> .
<b>RequestVersion</b>	<b>t:RequestServerVersion</b> ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the <b>Subscribe</b> operation request.

### 3.1.4.3.1.2 tns:SubscribeSoapOut Message

The **SubscribeSoapOut** WSDL message specifies the server response to the **Subscribe** operation request to subscribe to event notifications.

```
<wsdl:message name="SubscribeSoapOut">
  <wsdl:part name="SubscribeResult" element="tns:SubscribeResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **SubscribeSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/Subscribe>.

The parts of the **SubscribeSoapOut** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>SubscribeResult</b>	<b>tns:SubscribeResponse</b> (section <a href="#">3.1.4.3.2.2</a> )	Specifies the SOAP body of the response message.
<b>ServerVersion</b>	<b>t:ServerVersionInfo</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.12)	Specifies a SOAP header that identifies the schema version for the <b>Subscribe</b> operation (section <a href="#">3.1.4.3</a> ) message.

### 3.1.4.3.2 Elements

The following table lists the XML schema element definitions that are specific to the **Subscribe** operation.

Element name	Description
<b>Subscribe</b>	Specifies the properties that are used to create subscriptions.
<b>SubscribeResponse</b>	Specifies a response to a <b>Subscribe</b> operation request.

### 3.1.4.3.2.1 Subscribe Element

The **Subscribe** element specifies the properties that are used to create subscriptions.

```
<xs:element name="Subscribe"
  type="m:SubscribeType"
 />
```

### 3.1.4.3.2.2 SubscribeResponse Element

The **SubscribeResponse** element specifies a response to a **Subscribe** operation request.

```
<xs:element name="SubscribeResponse"
  type="m:SubscribeResponseType"
 />
```

### 3.1.4.3.3 Complex Types

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

Complex type name	Description
<b>SubscribeResponseType</b>	Specifies a response to a <b>Subscribe</b> operation.
<b>SubscribeType</b>	Specifies a request to subscribe to notifications of events on mailboxes.
<b>BaseSubscriptionRequestType</b>	Specifies the base class for descendant classes that form the notification requests.
<b>NonEmptyArrayOfNotificationEventTypesType</b>	Specifies an array of event notification types that are used to create a subscription.
<b>PullSubscriptionRequestType</b>	Specifies a subscription to a pull-based event notification subscription.
<b>PushSubscriptionRequestType</b>	Specifies a subscription to a push-based event notification.

#### 3.1.4.3.3.1 m:SubscribeResponseType Complex Type

The **SubscribeResponseType** complex type specifies a response to a **Subscribe** operation request. The **SubscribeResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

```
<xs:complexType name="SubscribeResponseType">
```

```

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

```

### 3.1.4.3.3.2 m:SubscribeType Complex Type

The **SubscribeType** complex type specifies a request to subscribe to notifications of events on mailboxes. The **SubscribeType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```

<xs:complexType name="SubscribeType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:choice>
        <xs:element name="PullSubscriptionRequest"
          type="t:PullSubscriptionRequestType"
        />
        <xs:element name="PushSubscriptionRequest"
          type="t:PushSubscriptionRequestType"
        />
        <xs:element name="StreamingSubscriptionRequest"
          type="t:StreamingSubscriptionRequestType"
        />
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists and describes the child elements of the **SubscribeType** complex type.

Element name	Type	Description
<b>PullSubscriptionRequest</b>	<b>t:PullSubscriptionRequestType</b> (section <a href="#">3.1.4.3.3.5</a> )	Specifies a pull subscription type.
<b>PushSubscriptionRequest</b>	<b>t:PushSubscriptionRequestType</b> (section <a href="#">3.1.4.3.3.6</a> )	Specifies a push subscription type.
<b>StreamingSubscriptionRequest</b>	<b>t:StreamingSubscriptionRequestType</b> (section <a href="#">3.1.4.2.3.3</a> )	Specifies a subscription to a streaming event notification. <a href="#">&lt;9&gt;</a>

### 3.1.4.3.3.3 t:BaseSubscriptionRequestType Complex Type

The **BaseSubscriptionRequestType** complex type specifies the base class for descendant classes that form the notification requests.

```

<xs:complexType name="BaseSubscriptionRequestType"
  abstract="true"
>
  <xs:sequence>
    <xs:element name="FolderIds"
      type="t:NonEmptyArrayOfBaseFolderIdsType"
    />
  </xs:sequence>
</xs:complexType>

```



```

        minOccurs="0"
      />
    <xs:element name="EventTypes"
      type="t:NonEmptyArrayOfNotificationEventTypesType"
    />
    <xs:element name="Watermark"
      type="t:WatermarkType"
      minOccurs="0"
    />
  </xs:sequence>
  <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
</xs:complexType>

```

The following table lists and describes the child elements of the **BaseSubscriptionRequestType** complex type.

Element name	Type	Description
<b>FolderIds</b>	<b>t:NonEmptyArrayOfBaseFolderIdsType</b> ( <a href="#">[MS-OXWSFOLD]</a> section 3.1.4.6.3.3)	Specifies an array of folder identifiers that are used to identify folders to be monitored for event notifications.
<b>EventTypes</b>	<b>t:NonEmptyArrayOfNotificationEventTypesType</b> (section <a href="#">3.1.4.3.3.4</a> )	Specifies the monitored event types for a notification subscription.
<b>Watermark</b>	<b>t:WatermarkType</b> (section <a href="#">2.2.5.1</a> )	Specifies an event bookmark in the mailbox event queue.

The following table lists and describes the attributes of the **BaseSubscriptionRequestType** complex type.

Attribute Name	Type	Description
<b>SubscribeToAllFolders</b>	<b>1. xs:boolean</b> <a href="#">[XMLSCHEMA2]</a>	If <b>true</b> , identifies that this subscription apply to all folders and <b>FolderIds</b> MUST be null or empty. <a href="#">&lt;10&gt;</a>

### 3.1.4.3.3.4 t:NonEmptyArrayOfNotificationEventTypesType Complex Type

The **NonEmptyArrayOfNotificationEventTypesType** complex type specifies an array of event notification types that are used to create a subscription. [<11>](#)

```

<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice
    minOccurs="unbounded"
  >
    <xs:element name="EventType"
      type="t:NotificationEventTypeType"
    />
  </xs:choice>
</xs:complexType>

```

The following table lists and describes the child elements of the **NonEmptyArrayOfNotificationEventTypesType** complex type.

Element name	Type	Description
<b>EventType</b>	<b>t:NotificationEventTypeType</b> (section <a href="#">3.1.4.3.4.1</a> )	Specifies a requested event notification type that is used to create a subscription.

### 3.1.4.3.3.5 t:PullSubscriptionRequestType Complex Type

The **PullSubscriptionRequestType** complex type specifies a subscription to a pull-based event notification subscription. The **PullSubscriptionRequestType** complex type extends the **BaseSubscriptionRequestType** complex type, as specified in section [3.1.4.3.3.3](#).

```
<xs:complexType name="PullSubscriptionRequestType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseSubscriptionRequestType"
    >
      <xs:sequence>
        <xs:element name="Timeout"
          type="t:SubscriptionTimeoutType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **PullSubscriptionRequestType** complex type.

Element name	Type	Description
<b>Timeout</b>	<b>t:SubscriptionTimeoutType</b> (section <a href="#">3.1.4.3.4.2</a> )	Specifies the amount of time that a subscription can remain idle without a <b>GetEvents</b> request from the client.

### 3.1.4.3.3.6 t:PushSubscriptionRequestType Complex Type

The **PushSubscriptionRequestType** complex type specifies a subscription to a push-based event notification. The **PushSubscriptionRequestType** complex type extends the **BaseSubscriptionRequestType** complex type, as specified in section [3.1.4.3.3.3](#).

```
<xs:complexType name="PushSubscriptionRequestType">
  <xs:complexContent>
    <xs:extension base="t:BaseSubscriptionRequestType">
      <xs:sequence>
        <xs:element name="StatusFrequency" type="t:SubscriptionStatusFrequencyType"/>
        <xs:element name="URL" type="xs:string"/>
        <xs:element name="CallerData" type="xs:string" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The following table lists and describes the child elements of the **PushSubscriptionRequestType** complex type.

Element name	Type	Description
<b>StatusFrequency</b>	<b>t:SubscriptionStatusFrequencyType</b> (section <a href="#">3.1.4.3.4.3</a> )	Specifies the interval, specified in minutes, at which push notification status will be sent to the client when the subscription is idle.
<b>URL</b>	<b>xs:string</b> <a href="#">[XMLSCHEMA2]</a>	Specifies the location of the client Web service for push notifications.
<b>CallerData</b>	<b>xs:string</b>	Specifies information about the caller. <a href="#">&lt;12&gt;</a>

### 3.1.4.3.4 Simple Types

The following table lists the XML schema simple type definitions that are specific to the **Subscribe** operation.

Simple type name	Description
<b>NotificationEventTypeType</b>	Specifies the event types that are reported in a notification.
<b>SubscriptionTimeoutType</b>	Specifies the amount of time that a subscription can remain idle without a <b>GetEvents</b> operation request from the client.
<b>SubscriptionStatusFrequencyType</b>	Specifies the interval, in minutes, at which push notification status will be sent to the client.

#### 3.1.4.3.4.1 t:NotificationEventTypeType Simple Type

The **NotificationEventTypeType** simple type specifies the event types that are reported in a notification.

```
<xs:simpleType name="NotificationEventTypeType">
  <xs:restriction>
    <xs:enumeration
      value="CopiedEvent"
    />
    <xs:enumeration
      value="CreatedEvent"
    />
    <xs:enumeration
      value="DeletedEvent"
    />
    <xs:enumeration
      value="ModifiedEvent"
    />
    <xs:enumeration
      value="MovedEvent"
    />
    <xs:enumeration
      value="NewMailEvent"
    />
    <xs:enumeration
      value="FreeBusyChangedEvent"
    />
  </xs:restriction>
</xs:simpleType>
```

The following table lists and describes the values that are defined by the **NotificationEventType** simple type.

Value name	Meaning
<b>CopiedEvent</b>	The notification is a copied event.
<b>CreatedEvent</b>	The notification is a created event.
<b>DeletedEvent</b>	The notification is a deleted event.
<b>ModifiedEvent</b>	The notification is a modified event.
<b>MovedEvent</b>	The notification is a moved event.
<b>NewMailEvent</b>	The notification is a new mail event.
<b>FreeBusyChangedEvent</b>	The notification is a free/busy changed event. <a href="#">&lt;13&gt;</a>

#### 3.1.4.3.4.2 t:SubscriptionTimeoutType Simple Type

The **SubscriptionTimeoutType** simple type specifies the duration that a subscription can remain idle without a **GetEvents** operation request from the client.

```
<xs:simpleType name="SubscriptionTimeoutType">
  <xs:restriction
    base="xs:int"
  >
    <xs:minInclusive
      value="1"
    />
    <xs:maxInclusive
      value="1440"
    />
  </xs:restriction>
</xs:simpleType>
```

#### 3.1.4.3.4.3 t:SubscriptionStatusFrequencyType Simple Type

The **SubscriptionStatusFrequencyType** simple type specifies the interval, in minutes, at which push notification status will be sent to the client when the subscription is idle.

```
<xs:simpleType name="SubscriptionStatusFrequencyType">
  <xs:restriction
    base="xs:int"
  >
    <xs:minInclusive
      value="1"
    />
    <xs:maxInclusive
      value="1440"
    />
  </xs:restriction>
</xs:simpleType>
```

### 3.1.4.4 Unsubscribe

The **Unsubscribe** operation removes a pull or streaming subscription to notifications of events on mailboxes. If other subscription types are specified, a **ResponseCode** ([\[MS-OXWSCDATA\]](#) section 2.2.5.24) element with **ErrorInvalidSubscription** will be returned. <14>

The following is the WSDL port type specification for the **Unsubscribe** operation.

```
<wsdl:operation name="Unsubscribe">
  <wsdl:input message="tns:UnsubscribeSoapIn" />
  <wsdl:output message="tns:UnsubscribeSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification for the **Unsubscribe** operation.

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

#### 3.1.4.4.1 Messages

The following table lists the WSDL message definitions are specific to the **Unsubscribe** operation.

Message name	Description
<b>UnsubscribeSoapIn</b>	Specifies the <b>Unsubscribe</b> operation WSDL message.
<b>UnsubscribeSoapOut</b>	Specifies the <b>Unsubscribe</b> operation response WSDL message.

##### 3.1.4.4.1.1 tns:UnsubscribeSoapIn Message

The **UnsubscribeSoapIn** WSDL message specifies the **Unsubscribe** operation request to remove a subscription to an event notification.

```
<wsdl:message name="UnsubscribeSoapIn">
  <wsdl:part name="request" element="tns:Unsubscribe" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

The **UnsubscribeSoapIn** WSDL message is the input message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/Unsubscribe>.

The parts of the **UnsubscribeSoapIn** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>request</b>	<b>tns:Unsubscribe</b> (section <a href="#">3.1.4.4.2.1</a> )	Specifies the SOAP body of the request.
<b>Impersonation</b>	<b>t:ExchangeImpersonation</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
<b>MailboxCulture</b>	<b>t:MailboxCulture</b> ([MS-OXWSCDATA] section 2.2.3.7)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in <a href="#">[RFC3066]</a> .
<b>RequestVersion</b>	<b>t:RequestServerVersion</b> ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the <b>Unsubscribe</b> operation request.

### 3.1.4.4.1.2 tns:UnsubscribeSoapOut Message

The **UnsubscribeSoapOut** WSDL message specifies the server response to the **Unsubscribe** operation request to remove a subscription to an event notification.

```
<wsdl:message name="UnsubscribeSoapOut">
  <wsdl:part name="UnsubscribeResult" element="tns:UnsubscribeResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **UnsubscribeSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/Unsubscribe>.

The parts of the **UnsubscribeSoapOut** WSDL message are listed and described in the following table.

Part name	Element/type	Description
<b>UnsubscribeResult</b>	<b>tns:UnsubscribeResponse</b> (section <a href="#">3.1.4.4.2.2</a> )	Specifies the SOAP body of the response message.
<b>ServerVersion</b>	<b>t:ServerVersionInfo</b> ( <a href="#">[MS-OXWSCDATA]</a> section 2.2.3.12)	Specifies a SOAP header that identifies the schema version for the <b>UnsubscribeSoapOut</b> message.

### 3.1.4.4.2 Elements

The following table lists the XML schema element definitions that are specific to the **Unsubscribe** operation.

Element name	Description
<b>Unsubscribe</b>	Specifies the properties used to end a subscription.
<b>UnsubscribeResponse</b>	Specifies a response to an <b>Unsubscribe</b> operation request.

#### 3.1.4.4.2.1 Unsubscribe Element

The **Unsubscribe** element specifies the properties used to end a subscription.

```
<xs:element name="Unsubscribe"
  type="m:UnsubscribeType"
  />
```

### 3.1.4.4.2 UnsubscribeResponse Element

The **UnsubscribeResponse** element specifies a response to an **Unsubscribe** operation request.

```
<xs:element name="UnsubscribeResponse"
  type="m:UnsubscribeResponseType"
  />
```

### 3.1.4.4.3 Complex Types

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

Complex type name	Description
<b>UnsubscribeResponseType</b>	Specifies a response to an <b>Unsubscribe</b> operation request.
<b>UnsubscribeType</b>	Specifies a request to end a pull notification subscription.

#### 3.1.4.4.3.1 m:UnsubscribeResponseType Complex Type

The **UnsubscribeResponseType** complex type specifies a response to an **Unsubscribe** operation request. The **UnsubscribeResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

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

#### 3.1.4.4.3.2 m:UnsubscribeType Complex Type

The **UnsubscribeType** complex type specifies a request to end a notification subscription. The **UnsubscribeType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="UnsubscribeType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
      >
    <xs:sequence>
      <xs:element name="SubscriptionId"
        type="t:SubscriptionIdType"
        />
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>
```

```
</xs:extension>  
</xs:complexContent>  
</xs:complexType>
```

The following table lists and describes the child elements of the **UnsubscribeType** complex type.

Element name	Type	Description
<b>SubscriptionId</b>	<b>t:SubscriptionIdType</b> (section <a href="#">2.2.5.2</a> )	Specifies the subscription identifier of the subscription to be canceled.

### 3.1.5 Timer Events

None.

### 3.1.6 Other Local Events

None.



## 4 Protocol Examples

None.

## **5 Security**

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

None.

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

None.

## 6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality described in this document.

File name	Description	Section
MS-OXWSNTIF.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSNTIF-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	<a href="#">7.1</a>
MS-OXWSNTIF-types.xsd	Contains the XML schema type definitions that are used in this protocol.	<a href="#">7.2</a>

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

This section contains the contents of the MS-OXWSNTIF.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-OXWSNTIF-messages.xsd" />
      <xs:include schemaLocation="MS-OXWSCDATA-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"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="SubscribeSoapIn">
    <wsdl:part name="request" element="tns:Subscribe" />
    <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="SubscribeSoapOut">
    <wsdl:part name="SubscribeResult" element="tns:SubscribeResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:message name="UnsubscribeSoapIn">
    <wsdl:part name="request" element="tns:Unsubscribe" />
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
  </wsdl:message>
</wsdl:definitions>
```

```

</wsdl:message>
<wsdl:message name="UnsubscribeSoapOut">
  <wsdl:part name="UnsubscribeResult" element="tns:UnsubscribeResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="GetEventsSoapIn">
  <wsdl:part name="request" element="tns:GetEvents" />
  <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="GetEventsSoapOut">
  <wsdl:part name="GetEventsResult" element="tns:GetEventsResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="GetStreamingEventsSoapIn">
  <wsdl:part name="request" element="tns:GetStreamingEvents" />
  <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="GetStreamingEventsSoapOut">
  <wsdl:part name="GetStreamingEventsResult" element="tns:GetStreamingEventsResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:portType name="ExchangeServicePortType">
  <wsdl:operation name="Subscribe">
    <wsdl:input message="tns:SubscribeSoapIn" />
    <wsdl:output message="tns:SubscribeSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="Unsubscribe">
    <wsdl:input message="tns:UnsubscribeSoapIn" />
    <wsdl:output message="tns:UnsubscribeSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetEvents">
    <wsdl:input message="tns:GetEventsSoapIn" />
    <wsdl:output message="tns:GetEventsSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="GetStreamingEvents">
    <wsdl:input message="tns:GetStreamingEventsSoapIn" />
    <wsdl:output message="tns:GetStreamingEventsSoapOut" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
  <wsdl:documentation>
    <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim"/>
  </wsdl:documentation>
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="Subscribe">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/Subscribe" />
    <wsdl:input>
      <soap:header message="tns:SubscribeSoapIn" part="Impersonation"
use="literal"/>
      <soap:header message="tns:SubscribeSoapIn" part="MailboxCulture"
use="literal"/>
      <soap:header message="tns:SubscribeSoapIn" part="RequestVersion"
use="literal"/>
      <soap:body parts="request" use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="SubscribeResult" use="literal" />
      <soap:header message="tns:SubscribeSoapOut" part="ServerVersion"
use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="Unsubscribe">

```

```

        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/Unsubscribe" />
        <wsdl:input>
            <soap:header message="tns:UnsubscribeSoapIn" part="Impersonation"
use="literal"/>
            <soap:header message="tns:UnsubscribeSoapIn" part="MailboxCulture"
use="literal"/>
            <soap:header message="tns:UnsubscribeSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="UnsubscribeResult" use="literal" />
            <soap:header message="tns:UnsubscribeSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetEvents">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents" />
        <wsdl:input>
            <soap:header message="tns:GetEventsSoapIn" part="Impersonation"
use="literal"/>
            <soap:header message="tns:GetEventsSoapIn" part="MailboxCulture"
use="literal"/>
            <soap:header message="tns:GetEventsSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="GetEventsResult" use="literal" />
            <soap:header message="tns:GetEventsSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetStreamingEvents">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents" />
        <wsdl:input>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="Impersonation"
use="literal"/>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="MailboxCulture"
use="literal"/>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="GetStreamingEventsResult" use="literal" />
            <soap:header message="tns:GetStreamingEventsSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>

```

## 7 Appendix B: Full XML Schema

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

Schema name	Prefix	Section
Messages schema	m:	<a href="#">7.1</a>
Types schema	t:	<a href="#">7.2</a>

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

### 7.1 Messages Schema

This section provides the contents of the MS-OXWSNTIF-messages.xsd file and information about additional files that this schema file requires to operate correctly.

MS-OXWSNTIF-messages.xsd includes the files listed in the following table. To operate correctly, these files have to be present in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-messages.xsd	<a href="#">[MS-OXWSCDATA]</a> section 7.1

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2016" id="messages">
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSNTIF-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:complexType name="GetEventsType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
          <xs:element name="Watermark" type="t:WatermarkType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetEvents" type="m:GetEventsType"/>
  <xs:complexType name="GetEventsResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="Notification" type="t:NotificationType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="GetEventsResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="GetEventsResponse" type="m:GetEventsResponseType"/>
</xs:schema>
```

```

<xs:complexType name="SubscribeType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:choice>
        <xs:element name="PullSubscriptionRequest" type="t:PullSubscriptionRequestType"/>
        <xs:element name="PushSubscriptionRequest" type="t:PushSubscriptionRequestType"/>
        <xs:element name="StreamingSubscriptionRequest"
type="t:StreamingSubscriptionRequestType"/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetStreamingEventsType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="SubscriptionIds" type="t:NonEmptyArrayOfSubscriptionIdsType"/>
        <xs:element name="ConnectionTimeout"
type="t:StreamingSubscriptionConnectionTimeoutType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="GetStreamingEvents" type="m:GetStreamingEventsType"/>
<!-- GetStreamingEvents response -->
<xs:complexType name="GetStreamingEventsResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="Notifications" type="t:NonEmptyArrayOfNotificationsType"
minOccurs="0"/>
        <xs:element name="ErrorSubscriptionIds" type="t:NonEmptyArrayOfSubscriptionIdsType"
minOccurs="0"/>
        <xs:element name="ConnectionStatus" type="t:ConnectionStatusType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetStreamingEventsResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="GetStreamingEventsResponse" type="m:GetStreamingEventsResponseType"/>
<xs:element name="Subscribe" type="m:SubscribeType"/>
<xs:complexType name="SubscribeResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="SubscriptionId" type="t:SubscriptionIdType" minOccurs="0"/>
        <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SubscribeResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SubscribeResponse" type="m:SubscribeResponseType"/>
<xs:complexType name="UnsubscribeType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>

```

```

</xs:complexType>
<xs:element name="Unsubscribe" type="m:UnsubscribeType"/>
<xs:complexType name="UnsubscribeResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="UnsubscribeResponse" type="m:UnsubscribeResponseType"/>
</xs:schema>

```

## 7.2 Types Schema

This section provides the contents of the MS-OXWSNTIF-types.xsd file and information about additional files that this schema file requires to operate correctly.

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

File name	Defining specification
MS-OXWSCDATA-types.xsd	<a href="#">[MS-OXWSCDATA]</a> section 7.2

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
  elementFormDefault="qualified" version="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:complexType name="BaseNotificationEventType">
    <xs:sequence>
      <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BaseObjectChangedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseNotificationEventType">
        <xs:sequence>
          <xs:element name="TimeStamp" type="xs:dateTime"/>
          <xs:choice>
            <xs:element name="FolderId" type="t:FolderIdType"/>
            <xs:element name="ItemId" type="t:ItemIdType"/>
          </xs:choice>
          <xs:element name="ParentFolderId" type="t:FolderIdType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="BaseSubscriptionRequestType" abstract="true">
    <xs:sequence>
      <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType" minOccurs="0"/>
      <xs:element name="EventTypes" type="t:NonEmptyArrayOfNotificationEventTypesType"/>
      <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
  </xs:complexType>
  <xs:complexType name="ModifiedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseObjectChangedEventType">
        <xs:sequence>
          <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```



```

    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MovedCopiedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseObjectChangedEventType">
        <xs:sequence>
          <xs:choice>
            <xs:element name="OldFolderId" type="t:FolderIdType"/>
            <xs:element name="OldItemId" type="t:ItemIdType"/>
          </xs:choice>
          <xs:element name="OldParentFolderId" type="t:FolderIdType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
    <xs:choice maxOccurs="unbounded">
      <xs:element name="EventType" type="t:NotificationEventTypeType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="NotificationType">
    <xs:sequence>
      <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
      <xs:element name="PreviousWatermark" type="t:WatermarkType" minOccurs="0"/>
      <xs:element name="MoreEvents" type="xs:boolean" minOccurs="0"/>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="CopiedEvent" type="t:MovedCopiedEventType"/>
        <xs:element name="CreatedEvent" type="t:BaseObjectChangedEventType"/>
        <xs:element name="DeletedEvent" type="t:BaseObjectChangedEventType"/>
        <xs:element name="ModifiedEvent" type="t:ModifiedEventType"/>
        <xs:element name="MovedEvent" type="t:MovedCopiedEventType"/>
        <xs:element name="NewMailEvent" type="t:BaseObjectChangedEventType"/>
        <xs:element name="StatusEvent" type="t:BaseNotificationEventType"/>
        <xs:element name="FreeBusyChangedEvent" type="t:BaseObjectChangedEventType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="NotificationEventTypeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CopiedEvent"/>
      <xs:enumeration value="CreatedEvent"/>
      <xs:enumeration value="DeletedEvent"/>
      <xs:enumeration value="ModifiedEvent"/>
      <xs:enumeration value="MovedEvent"/>
      <xs:enumeration value="NewMailEvent"/>
      <xs:enumeration value="FreeBusyChangedEvent"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="PushSubscriptionRequestType">
    <xs:complexContent>
      <xs:extension base="t:BaseSubscriptionRequestType">
        <xs:sequence>
          <xs:element name="StatusFrequency" type="t:SubscriptionStatusFrequencyType"/>
          <xs:element name="URL" type="xs:string"/>
          <xs:element name="CallerData" type="xs:string" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="PullSubscriptionRequestType">
    <xs:complexContent>
      <xs:extension base="t:BaseSubscriptionRequestType">
        <xs:sequence>
          <xs:element name="Timeout" type="t:SubscriptionTimeoutType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

<xs:simpleType name="SubscriptionIdType">
  <xs:restriction base="t:NonEmptyStringType"/>
</xs:simpleType>
<xs:simpleType name="WatermarkType">
  <xs:restriction base="t:NonEmptyStringType"/>
</xs:simpleType>
<xs:simpleType name="SubscriptionTimeoutType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="1440"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SubscriptionStatusFrequencyType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="1440"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="StreamingSubscriptionRequestType">
  <xs:sequence>
    <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType" minOccurs="0"/>
    <xs:element name="EventTypes" type="t:NonEmptyArrayOfNotificationEventTypesType"/>
  </xs:sequence>
  <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfSubscriptionIdsType">
  <xs:sequence>
    <xs:element name="SubscriptionId" type="t:SubscriptionIdType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="NonEmptyArrayOfNotificationsType">
  <xs:sequence>
    <xs:element name="Notification" type="t:NotificationType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="StreamingSubscriptionConnectionTimeoutType">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="30"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ConnectionStatusType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OK"/>
    <xs:enumeration value="Closed"/>
  </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 released service packs.

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

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification 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 2.2.4.2](#): Exchange 2007 and Exchange 2010 do not support the **GetStreamingEventsResponseMessageType** complex type. This type was introduced in Microsoft Exchange Server 2010 Service Pack 1 (SP1).

<2> [Section 2.2.4.4](#): Exchange 2010, Exchange 2013, and Exchange 2016 do not include the **minOccurs** value.

<3> [Section 2.2.4.6](#): In Exchange 2010, Exchange 2013, and Exchange 2016, the **UnreadCount** element includes a **maxOccurs** value of 1.

<4> [Section 2.2.4.8](#): Exchange 2010, Exchange 2013, and Exchange 2016 do not include the **minOccurs** value for the element.

<5> [Section 2.2.4.8](#): Exchange 2010, Exchange 2013, and Exchange 2016 do not include the **minOccurs** value for this element.

<6> [Section 2.2.4.8](#): In Exchange 2010, Exchange 2013, and Exchange 2016 the **xs:choice** element includes a **minOccurs** value of 1.

<7> [Section 2.2.4.8](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 do not include the **FreeBusyChangedEvent** element.

<8> [Section 3.1.4.2](#): Exchange 2007 and Exchange 2010 do not include the **GetStreamingEvents** operation. The **GetStreamingEvents** operation was introduced in Exchange 2010 SP1 .

<9> [Section 3.1.4.3.3.2](#): Exchange 2007 and Exchange 2010 do not include the **StreamingSubscriptionRequest** element. The **StreamingSubscriptionRequest** operation was introduced in Exchange 2010 SP1.

<10> [Section 3.1.4.3.3.3](#): Exchange 2010 does not include the **SubscribeToAllFolders** attribute.

<11> [Section 3.1.4.3.3.4](#): In Exchange 2010, Exchange 2013, and Exchange 2016, the **xs:choice** element has a **minOccurs** value of 1.

<12> [Section 3.1.4.3.3.6](#): Exchange 2007 and Exchange 2010 do not support the **CallerData** element.

<13> [Section 3.1.4.3.4.1](#): Exchange 2007, Exchange 2010, Exchange 2013, and Exchange 2016 do not include the **FreeBusyChangedEvent** enumeration value.

<14> [Section 3.1.4.4](#): Exchange 2007 returns **ErrorInvalidPullSubscriptionId**.

## 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

### A

Abstract data model  
[server](#) 18  
[Applicability](#) 8  
[Attribute groups](#) 17  
[Attributes](#) 17

### C

[Capability negotiation](#) 8  
[Change tracking](#) 52  
[Complex types](#) 10  
[m:GetEventsResponseMessageType Complex Type](#)  
10  
[m:GetStreamingEventsResponseMessageType  
Complex Type](#) 11  
[m:SubscribeResponseMessageType Complex Type](#)  
11  
[t:BaseNotificationEventType Complex Type](#) 12  
[t:BaseObjectChangedEventType Complex Type](#) 12  
[t:ModifiedEventType Complex Type](#) 13  
[t:MovedCopiedEventType Complex Type](#) 14  
[t:NotificationType Complex Type](#) 15

### D

Data model - abstract  
[server](#) 18

### E

Events  
[local - server](#) 39  
[timer - server](#) 39  
[Examples](#) 40

### F

[Fields - vendor-extensible](#) 8  
[Full WSDL](#) 42  
[Full XML schema](#) 45  
[Messages Schema](#) 45  
[Types Schema](#) 47

### G

[Glossary](#) 5  
[Groups](#) 17

### I

[Implementer - security considerations](#) 41  
[Index of security parameters](#) 41  
[Informative references](#) 7  
Initialization  
[server](#) 18  
[Introduction](#) 5

### L

Local events

[server](#) 39

### M

[m:GetEventsResponseMessageType Complex Type  
complex type](#) 10  
[m:GetStreamingEventsResponseMessageType  
Complex Type complex type](#) 11  
[m:SubscribeResponseMessageType Complex Type  
complex type](#) 11  
Message processing  
[server](#) 18  
Messages  
[attribute groups](#) 17  
[attributes](#) 17  
[complex types](#) 10  
[elements](#) 9  
[enumerated](#) 9  
[groups](#) 17  
[m:GetEventsResponseMessageType Complex Type  
complex type](#) 10  
[m:GetStreamingEventsResponseMessageType  
Complex Type complex type](#) 11  
[m:SubscribeResponseMessageType Complex Type  
complex type](#) 11  
[namespaces](#) 9  
[simple types](#) 16  
[syntax](#) 9  
[t:BaseNotificationEventType Complex Type  
complex type](#) 12  
[t:BaseObjectChangedEventType Complex Type  
complex type](#) 12  
[t:ModifiedEventType Complex Type complex type](#)  
13  
[t:MovedCopiedEventType Complex Type complex  
type](#) 14  
[t:NotificationType Complex Type complex type](#) 15  
[t:SubscriptionIdType Simple Type simple type](#) 16  
[t:WatermarkType Simple Type simple type](#) 16  
[transport](#) 9

### N

[Namespaces](#) 9  
[Normative references](#) 6

### O

Operations  
[GetEvents](#) 18  
[GetStreamingEvents](#) 22  
[Subscribe](#) 28  
[Unsubscribe](#) 36  
[Overview \(synopsis\)](#) 7

### P

[Parameters - security index](#) 41  
[Preconditions](#) 8  
[Prerequisites](#) 8  
[Product behavior](#) 50  
Protocol Details

[overview](#) 18

[WSDL](#) 42

## R

[References](#) 6

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

## X

[XML schema](#) 45

[Messages Schema](#) 45

[Types Schema](#) 47

## S

Security

[implementer considerations](#) 41

[parameter index](#) 41

Sequencing rules

[server](#) 18

Server

[abstract data model](#) 18

[GetEvents operation](#) 18

[GetStreamingEvents operation](#) 22

[initialization](#) 18

[local events](#) 39

[message processing](#) 18

[sequencing rules](#) 18

[Subscribe operation](#) 28

[timer events](#) 39

[timers](#) 18

[Unsubscribe operation](#) 36

[Simple types](#) 16

[t:SubscriptionIdType Simple Type](#) 16

[t:WatermarkType Simple Type](#) 16

[Standards assignments](#) 8

Syntax

[messages - overview](#) 9

## T

[t:BaseNotificationEventType Complex Type complex type](#) 12

[t:BaseObjectChangedEventType Complex Type complex type](#) 12

[t:ModifiedEventType Complex Type complex type](#) 13

[t:MovedCopiedEventType Complex Type complex type](#) 14

[t:NotificationType Complex Type complex type](#) 15

[t:SubscriptionIdType Simple Type simple type](#) 16

[t:WatermarkType Simple Type simple type](#) 16

Timer events

[server](#) 39

Timers

[server](#) 18

[Tracking changes](#) 52

[Transport](#) 9

Types

[complex](#) 10

[simple](#) 16

## V

[Vendor-extensible fields](#) 8

[Versioning](#) 8

## W