

# [MS-OXWSNTIF]: Notifications Web Service Protocol Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). 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.
- **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.

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

## Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.1.0	Minor	Updated the technical content.
02/10/2010	1.1.0	None	Version 1.1.0 release
05/05/2010	2.0.0	Major	Updated and revised the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 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 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Common Message Syntax	9
2.2.1 Namespaces	9
2.2.2 Simple Types	9
2.2.2.1 t:WatermarkType Simple Type	9
2.2.2.2 t:SubscriptionIdType Simple Type	10
2.2.3 Complex Types	10
2.2.3.1 m:GetEventsResponseMessageType Complex Type	10
2.2.3.2 m:GetStreamingEventsResponseMessageType Complex Type	11
2.2.3.3 m:SubscribeResponseMessageType Complex Type	11
2.2.3.4 t:BaseNotificationEventType Complex Type	12
2.2.3.5 t:BaseObjectChangedEventType Complex Type	12
2.2.3.6 t:ModifiedEventType Complex Type	13
2.2.3.7 t:MovedCopiedEventType Complex Type	14
2.2.3.8 t:NotificationType Complex Type	14
2.2.4 Elements	16
2.2.5 Attributes	16
2.2.6 Groups	16
2.2.7 Attribute Groups	16
2.2.8 Messages	16
<b>3 Protocol Details</b>	<b>17</b>
3.1 ExchangeServicePortType Server Details	17
3.1.1 Abstract Data Model	17
3.1.2 Timers	17
3.1.3 Initialization	17
3.1.4 Message Processing Events and Sequencing	17
3.1.4.1 GetEvents	17
3.1.4.1.1 Complex Types	18
3.1.4.1.1.1 m:GetEventsResponseType Complex Type	18
3.1.4.1.1.2 m:GetEventsType Complex Type	18
3.1.4.1.2 Elements	19
3.1.4.1.2.1 GetEvents Element	19
3.1.4.1.2.2 GetEventsResponse Element	19
3.1.4.1.3 Messages	19
3.1.4.1.3.1 tns:GetEventsSoapIn Message	19

3.1.4.1.3.2	tns:GetEventsSoapOut Message .....	20
3.1.4.2	GetStreamingEvents .....	20
3.1.4.2.1	Simple Types .....	20
3.1.4.2.1.1	t:StreamingSubscriptionConnectionTimeoutType Simple Type .....	20
3.1.4.2.1.2	t:ConnectionStatusType Simple Type .....	21
3.1.4.2.2	Complex Types .....	21
3.1.4.2.2.1	m:GetStreamingEventsType Complex Type.....	21
3.1.4.2.2.2	m:GetStreamingEventsResponseType Complex Type .....	22
3.1.4.2.2.3	t:StreamingSubscriptionRequestType Complex Type .....	22
3.1.4.2.2.4	t:NonEmptyArrayOfSubscriptionIdsType Complex Type .....	23
3.1.4.2.2.5	t:NonEmptyArrayOfNotificationsType Complex Type .....	23
3.1.4.2.3	Elements.....	24
3.1.4.2.3.1	GetStreamingEvents Element.....	24
3.1.4.2.3.2	GetStreamingEventsResponse Element .....	24
3.1.4.2.4	Messages .....	24
3.1.4.2.4.1	tns:GetStreamingEventsSoapIn.....	24
3.1.4.2.4.2	tns:GetStreamingEventsSoapOut.....	25
3.1.4.3	Subscribe .....	25
3.1.4.3.1	Simple Types.....	25
3.1.4.3.1.1	t:NotificationEventTypeType Simple Type .....	25
3.1.4.3.1.2	t:SubscriptionTimeoutType Simple Type .....	26
3.1.4.3.1.3	t:SubscriptionStatusFrequencyType Simple Type .....	27
3.1.4.3.2	Complex Types .....	27
3.1.4.3.2.1	m:SubscribeResponseType Complex Type .....	27
3.1.4.3.2.2	m:SubscribeType Complex Type.....	27
3.1.4.3.2.3	t:BaseSubscriptionRequestType Complex Type.....	28
3.1.4.3.2.4	t:NonEmptyArrayOfNotificationEventTypesType Complex Type .....	29
3.1.4.3.2.5	t:PullSubscriptionRequestType Complex Type .....	29
3.1.4.3.2.6	t:PushSubscriptionRequestType Complex Type.....	30
3.1.4.3.3	Elements.....	30
3.1.4.3.3.1	Subscribe Element.....	30
3.1.4.3.3.2	SubscribeResponse Element.....	31
3.1.4.3.4	Messages.....	31
3.1.4.3.4.1	tns:SubscribeSoapIn Message.....	31
3.1.4.3.4.2	tns:SubscribeSoapOut Message.....	31
3.1.4.4	Unsubscribe .....	31
3.1.4.4.1	Complex Types .....	32
3.1.4.4.1.1	m:UnsubscribeResponseType Complex Type .....	32
3.1.4.4.1.2	m:UnsubscribeType Complex Type .....	32
3.1.4.4.2	Elements.....	32
3.1.4.4.2.1	Unsubscribe Element.....	33
3.1.4.4.2.2	UnsubscribeResponse Element .....	33
3.1.4.4.3	Messages .....	33
3.1.4.4.3.1	tns:UnsubscribeSoapIn Message .....	33
3.1.4.4.3.2	tns:UnsubscribeSoapOut Message .....	33
3.1.5	Timer Events .....	34
3.1.6	Other Local Events .....	34
3.2	Client Details.....	34
3.2.1	Abstract Data Model .....	34
3.2.2	Timers .....	34
3.2.3	Initialization .....	34
3.2.4	Message Processing Events and Sequencing .....	34
3.2.5	Timer Events .....	34

3.2.6 Other Local Events .....	34
<b>4 Protocol Examples .....</b>	<b>35</b>
<b>5 Security .....</b>	<b>36</b>
5.1 Security Considerations for Implementers .....	36
5.2 Index of Security Parameters .....	36
<b>6 Appendix A: Full WSDL .....</b>	<b>37</b>
6.1 WSDL .....	37
6.2 Types Schema .....	40
6.3 Messages Schema .....	43
<b>7 Appendix B: Product Behavior .....</b>	<b>46</b>
<b>8 Change Tracking .....</b>	<b>47</b>
<b>9 Index .....</b>	<b>51</b>

# 1 Introduction

This document specifies the Notifications Web Service protocol, which is responsible for pull notifications from the server. Clients use the SOAP protocol [\[SOAP1.1\]](#) to contact the notification service.

## 1.1 Glossary

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

**Hypertext Transfer Protocol (HTTP)**  
**Hypertext Transfer Protocol over Secure Socket Layers (HTTPS)**  
**mailbox**  
**SOAP body**  
**SOAP fault**  
**SOAP header**  
**SOAP message**  
**Web Services Description Language (WSDL)**  
**WSDL message**  
**WSDL port type**  
**XML**  
**XML namespace**  
**XML schema**

The following terms are specific to this document:

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

## 1.2 References

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

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

[MS-OXWSCDATA] Microsoft Corporation, "[Common Web Service Data Types](#)", July 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

[RFC2396] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>

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

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, BCP 47, 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", W3C Note, May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

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

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

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

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

### 1.2.2 Informative References

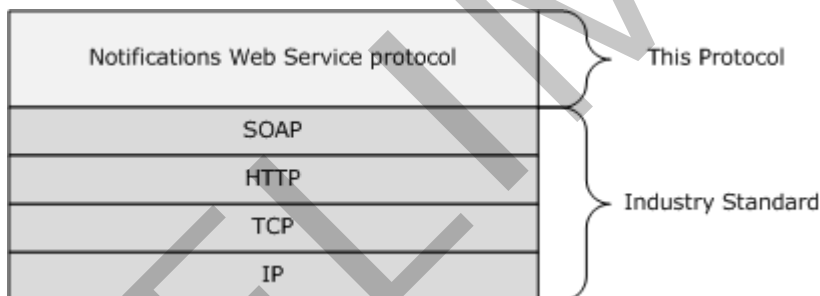
None.

### 1.3 Overview

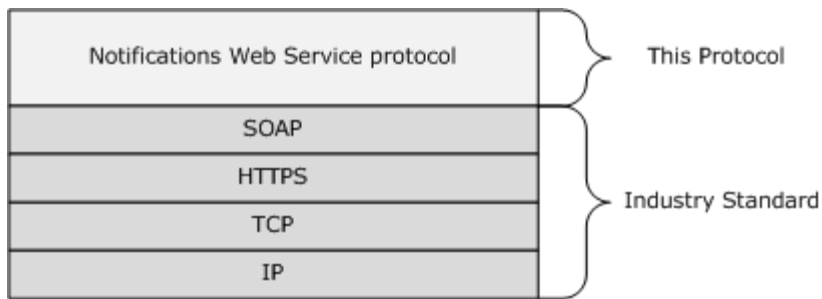
The Notifications Web Service protocol is used to enable clients to pull notifications from the server. Clients subscribe to updates and receive the updates through pull notifications, by which the client periodically requests information from the server.

### 1.4 Relationship to Other Protocols

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



**Figure 1: Notification Web Service protocol HTTP stack**



**Figure 2: Notification Web Service protocol HTTPS stack**

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

### 1.5 Prerequisites/Preconditions

None.

### 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 SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).
- **Capability Negotiation:** None.

### 1.8 Vendor Extensible Fields

None.

### 1.9 Standards Assignments

None.



## 2 Messages

### 2.1 Transport

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

### 2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema** as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and **Web Services Description Language (WSDL)** as defined in [\[WSDL\]](#).

#### 2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific XML namespaces prefix for each XML namespaces that is used, the choice of any particular XML namespaces 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>	[MS-OXWSNTIF]
s	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	<a href="#">[XMLSCHEMA1]</a>
targetNamespace	<a href="http://schemas.microsoft.com/exchange/services/2006/messages">http://schemas.microsoft.com/exchange/services/2006/messages</a>	[MS-OXWSNTIF]
wsdl	<a href="http://schemas.xmlsoap.org/wsdl/">http://schemas.xmlsoap.org/wsdl/</a>	<a href="#">[WSDL]</a>
t	<a href="http://schemas.microsoft.com/exchange/services/2006/types">http://schemas.microsoft.com/exchange/services/2006/types</a>	[MS-OXWSNTIF]

#### 2.2.2 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	Description
<a href="#">t:WatermarkType</a>	Specifies an event bookmark in the <b>mailbox</b> event queue.
<a href="#">t:SubscriptionIdType</a>	Specifies the identifier for a subscription.

##### 2.2.2.1 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>
```

### 2.2.2.2 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>
```

### 2.2.3 Complex Types

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

Complex Type	Description
<a href="#">m:GetEventsResponseMessageType</a>	Specifies the status and result of a single GetEvents request.
<a href="#">m:SubscribeResponseMessageType</a>	Specifies the status and result of a single Subscribe request.
<a href="#">t:ModifiedEventType</a>	Specifies an event in which an item or folder is modified.
<a href="#">t:MovedCopiedEventType</a>	Specifies a moved or copied event notification.
<a href="#">t:NotificationType</a>	Specifies information about the subscription and the events that have occurred since the last notification.

#### 2.2.3.1 m:GetEventsResponseMessageType Complex Type

The [GetEventsResponseMessageType](#) complex type specifies the status and result of a single GetEvents request.

```
<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>
```

Child Elements

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

### 2.2.3.2 m:GetStreamingEventsResponseMessageType Complex Type

The **[GetStreamingEventsResponseMessageType](#)** complex type specifies the status and result of a single **[GetStreamingEvents](#)** operation request. The **[GetStreamingEventsResponseMessageType](#)** complex type extends the **[ResponseMessageType](#)** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.3.52.

```
<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>
```

#### Child Elements

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

### 2.2.3.3 m:SubscribeResponseMessageType Complex Type

The **[SubscribeResponseMessageType](#)** complex type specifies the status and result of a single **[Subscribe](#)** operation (section [3.1.4.3](#)) request.

```

<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>

```

#### Child Elements

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

#### 2.2.3.4 t:BaseNotificationEventType Complex Type

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

```

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

```

#### Child Elements

Element	Type	Description
Watermark	<a href="#">t:WatermarkType</a>	Specifies an event bookmark in the mailbox event queue.

#### 2.2.3.5 t:BaseObjectChangedEventType Complex Type

The **BaseObjectChangedEventType** complex type specifies create, delete, and new mail events.

```

<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>

```

#### Child Elements

Element	Type	Description
TimeStamp	xs:dateTime	Specifies the time stamp of a changed item or folder mailbox event.
FolderId	<a href="#">t:FolderIdType</a>	Specifies a FolderIdType object that identifies the changed store item.
ItemId	<a href="#">t:ItemIdType</a>	Specifies an ItemIdType object that identifies the changed store item.
ParentFolderId	t:FolderIdType	Specifies the parent folder identifier of the changed item or folder.

#### 2.2.3.6 t:ModifiedEventType Complex Type

The [ModifiedEventType](#) complex type specifies an event in which an item or folder is modified.

```

<xs:complexType name="ModifiedEventType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseObjectChangedEventType"
    >
      <xs:sequence>
        <xs:element name="UnreadCount"
          type="xs:int"
          minOccurs="0"
          maxOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## Child Elements

Element	Type	Description
UnreadCount	xs:int	Specifies the count of unread items within a given folder.

### 2.2.3.7 t:MovedCopiedEventType Complex Type

The [MovedCopiedEventType](#) complex type specifies a moved or copied event notification.

```
<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>
```

## Child Elements

Element	Type	Description
OldFolderId	<a href="#">t:FolderIdType</a>	Specifies the old identifier for a copied or moved folder.
OldItemId	<a href="#">t:ItemIdType</a>	Specifies the old identifier for a copied or moved item.
OldParentFolderId	t:FolderIdType	Specifies the identifier of the original parent folder of an item or folder that was moved.

### 2.2.3.8 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>

```

#### Child Elements

Element	Type	Description
SubscriptionId	<a href="#">t:SubscriptionIdType</a>	Specifies the identifier for a subscription.
PreviousWatermark	<a href="#">t:WatermarkType</a>	Specifies the watermark of the latest event that was successfully communicated to the client for the subscription.
MoreEvents	xs:boolean	Specifies a value that indicates whether more events to be delivered to the client exist in the queue.
CopiedEvent	<a href="#">t:MovedCopiedEventType</a>	Specifies a copied event notification.

Element	Type	Description
CreatedEvent	<a href="#">t:BaseObjectChangedEventType</a>	Specifies a created event notification.
DeletedEvent	t:BaseObjectChangedEventType	Specifies a deleted event notification.
ModifiedEvent	<a href="#">t:ModifiedEventType</a>	Specifies a modified event notification.
MovedEvent	t:MovedCopiedEventType	Specifies a moved event notification.
NewMailEvent	t:BaseObjectChangedEventType	Specifies a new e-mail event notification.
StatusEvent	<a href="#">t:BaseNotificationEventType</a>	Specifies a status event notification.
FreeBusyChangedEvent	t:BaseObjectChangedEventType	Specifies an event in which an item's free/busy time has changed.

#### 2.2.4 Elements

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

#### 2.2.5 Attributes

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

#### 2.2.6 Groups

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

#### 2.2.7 Attribute Groups

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

#### 2.2.8 Messages

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



## 3 Protocol Details

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

### 3.1 ExchangeServicePortType Server Details

The Notifications Web Service protocol defines a single port type with four operations, as listed in the following table.

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

#### 3.1.1 Abstract Data Model

The Notifications Web Service protocol is a stateless protocol.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

None.

#### 3.1.4 Message Processing Events and Sequencing

This protocol includes the operations that are listed in the following table.

Operation	Description
<a href="#">GetEvents</a> (section <a href="#">3.1.4.1</a> )	Gets events for a pull notification subscription.
<a href="#">GetStreamingEvents</a> (section <a href="#">3.1.4.2</a> )	Gets events for a streaming notification subscription.
<a href="#">Subscribe</a> (section <a href="#">3.1.4.3</a> )	Creates a subscription to notifications of events on mailboxes.
<a href="#">Unsubscribe</a> (section <a href="#">3.1.4.4</a> )	Removes a subscription to notifications of events on mailboxes.

##### 3.1.4.1 GetEvents

The [GetEvents](#) operation gets events for a pull notification subscription.

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

</wsdl:operation>

#### Request

Message Format	Description
<a href="#">tns:GetEventsSoapIn</a>	Specifies the <b>SOAP message</b> that requests the events.

#### Response

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

### 3.1.4.1.1 Complex Types

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

#### 3.1.4.1.1.1 m:GetEventsResponseType Complex Type

The [GetEventsResponseType](#) complex type specifies a response to a [GetEvents](#) operation (section [3.1.4.1](#)) request.

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

#### 3.1.4.1.1.2 m:GetEventsType Complex Type

The **GetEventsType** complex type specifies a request to get events for a pull notification subscription.

```
<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>
```

## Child Elements

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

### 3.1.4.1.2 Elements

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

#### 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 (section [3.1.4.1](#)) request.

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

### 3.1.4.1.3 Messages

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

#### 3.1.4.1.3.1 tns:GetEventsSoapIn Message

The GetEventsSoapIn message contains four parts, as described in the following table.

Part Name	Element/Type	Description
request	<a href="#">tns:GetEvents</a>	Specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	Specifies the user whom the client application is impersonating.
MailboxCulture	<a href="#">t:MailboxCulture</a>	Specifies the culture to use for accessing the mailbox. The cultures are defined by <a href="#">[RFC3066]</a> .

Part Name	Element/Type	Description
RequestVersion	<a href="#">t:RequestServerVersion</a>	Specifies the schema version for the <b>GetEvents</b> operation (section <a href="#">3.1.4.1</a> ) request.

### 3.1.4.1.3.2 tns:GetEventsSoapOut Message

The GetEventsSoapOut message contains two parts, as described in the following table.

Part Name	Element/Type	Description
GetEventsResult	<a href="#">tns:GetEventsResponse</a>	Specifies the response message.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	Specifies the schema version for the GetEventsSoapOut message.

### 3.1.4.2 GetStreamingEvents

The **GetStreamingEvents** operation gets events for a streaming notification subscription.

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

Request

Message Format	Description
<a href="#">tns:GetStreamingEventsSoapIn</a>	Specifies the SOAP message that requests the events.

Response

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

#### 3.1.4.2.1 Simple Types

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

##### 3.1.4.2.1.1 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>

```

### 3.1.4.2.1.2 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>

```

Enumeration

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

Value	Description
OK	The connection is good.
Closed	The connection is closed.

### 3.1.4.2.2 Complex Types

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

#### 3.1.4.2.2.1 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 2.2.3.14.

```

<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>

```

#### Child Elements

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

#### 3.1.4.2.2.2 m:GetStreamingEventsResponseType Complex Type

The **GetStreamingEventsResponseType** complex type represents a response to a **GetStreamingEvents** operation (section [3.1.4.2](#)) request. The **GetStreamingEventsResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.3.15.

```

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

```

#### 3.1.4.2.2.3 t:StreamingSubscriptionRequestType Complex Type

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

```

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

```

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

#### Child Elements

Element	Type	Description
FolderIds	<a href="#">t:NonEmptyArrayOfBaseFolderIdsType</a>	Specifies an array of folder identifiers that are used to identify folders to monitor for event notifications.
EventTypes	<a href="#">t:NonEmptyArrayOfNotificationEventTypesType</a>	Specifies a collection of event notifications that are used to create a subscription.

#### 3.1.4.2.2.4 t:NonEmptyArrayOfSubscriptionIdsType Complex Type

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

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

#### Child Elements

Element	Type	Description
SubscriptionId	<a href="#">t:SubscriptionIdType</a>	Specifies the identifier for a subscription.

#### 3.1.4.2.2.5 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"  
      minOccurs="0"  
    />  
  </xs:sequence>  
</xs:complexType>
```

#### Child Elements

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

### 3.1.4.2.3 Elements

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

#### 3.1.4.2.3.1 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"
 />
```

#### 3.1.4.2.3.2 GetStreamingEventsResponse Element

The <GetStreamingEventsResponse> element specifies a response to a **GetStreamingEvents** operation (section [3.1.4.2](#)) request.

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

### 3.1.4.2.4 Messages

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

#### 3.1.4.2.4.1 tns:GetStreamingEventsSoapIn

The **GetStreamingEventsSoapIn** message contains four parts, as described in the following table.

Part name	Element/type	Description
request	<a href="#">tns:GetStreamingEvents</a>	Specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	Specifies the user whom the client application is impersonating.
MailboxCulture	<a href="#">t:MailboxCulture</a>	Specifies the culture to use for accessing the mailbox. The cultures are defined by <a href="#">[RFC3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	Specifies the schema version for the <b>Subscribe</b> operation (section <a href="#">3.1.4.3</a> ) request.



### 3.1.4.2.4.2 tns:GetStreamingEventsSoapOut

The **GetStreamingEventsSoapOut** message contains two parts, as described in the following table.

Part name	Element/type	Description
GetStreamingEventsResult	<GetStreamingEventsResponse> (section <a href="#">3.1.4.2.3.2</a> )	Specifies the response message.
ServerVersion	<ServerVersionInfo> (section <a href="#">2.2.4.8</a> )	Specifies the schema version for the <b>GetStreamingEvents</b> operation (section <a href="#">3.1.4.2</a> ) message.

### 3.1.4.3 Subscribe

The [Subscribe](#) operation creates a subscription to notifications of events on mailboxes in the mail server database.

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

Request

Message Format	Description
<a href="#">tns:SubscribeSoapIn</a>	Specifies the SOAP message that requests the subscription.

Response

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

### 3.1.4.3.1 Simple Types

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

#### 3.1.4.3.1.1 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:restriction>
</xs:simpleType>
```

```

<xs:enumeration
  value="DeletedEvent"
/>
<xs:enumeration
  value="ModifiedEvent"
/>
<xs:enumeration
  value="MovedEvent"
/>
<xs:enumeration
  value="NewMailEvent"
/>
<xs:enumeration
  value="FreeBusyChangedEvent"
/>
</xs:restriction>
</xs:simpleType>

```

## Enumeration

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

Value	Description
CopiedEvent	Identifies the notification as a copied event.
CreatedEvent	Identifies the notification as a created event.
DeletedEvent	Identifies the notification as a deleted event.
ModifiedEvent	Identifies the notification as a modified event.
MovedEvent	Identifies the notification as a moved event.
NewMailEvent	Identifies the notification as a new mail event.
FreeBusyChangedEvent	Identifies the notification as a free/busy changed event.

### 3.1.4.3.1.2 t:SubscriptionTimeoutType Simple Type

The **SubscriptionTimeoutType** simple type specifies the duration that a subscription can remain idle without a **GetEvents** operation (section [3.1.4.1](#)) 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>

```

### 3.1.4.3.1.3 t:SubscriptionStatusFrequencyType Simple Type

The [SubscriptionStatusFrequencyType](#) simple type specifies the frequency, 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>
```

### 3.1.4.3.2 Complex Types

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

#### 3.1.4.3.2.1 m:SubscribeResponseType Complex Type

The [SubscribeResponseType](#) complex type specifies a response to a [Subscribe](#) operation (section [3.1.4.3](#)) request.

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

#### 3.1.4.3.2.2 m:SubscribeType Complex Type

The **SubscribeType** complex type specifies a request to subscribe to notifications of events on mailboxes.

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

```

    <xs:element name="PushSubscriptionRequest"
      type="t:PushSubscriptionRequestType"
    />
    <xs:element name="StreamingSubscriptionRequest"
      type="t:StreamingSubscriptionRequestType"
    />
  </xs:choice>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

#### Child Elements

Element	Type	Description
PullSubscriptionRequest	<a href="#">t:PullSubscriptionRequestType</a>	Specifies a pull subscription type.
PushSubscriptionRequest	<a href="#">t:PushSubscriptionRequestType</a>	Specifies a push subscription type.
StreamingSubscriptionRequest	<a href="#">t:StreamingSubscriptionRequestType</a>	Specifies a subscription to a streaming event notification.

#### 3.1.4.3.2.3 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:element name="EventTypes"
      type="t:NonEmptyArrayOfNotificationEventTypesType"
    />
    <xs:element name="Watermark"
      type="t:WatermarkType"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>

```

#### Child Elements

Element	Type	Description
FolderIds	<a href="#">t:NonEmptyArrayOfBaseFolderIdsType</a>	Specifies an array of folder identifiers that are used to identify folders to monitor for event notifications.

Element	Type	Description
EventTypes	<a href="#">t:NonEmptyArrayOfNotificationEventTypesType</a>	Specifies the monitored event types for a notification subscription.
Watermark	<a href="#">t:WatermarkType</a>	Specifies an event bookmark in the mailbox event queue.

### 3.1.4.3.2.4 t:NonEmptyArrayOfNotificationEventTypesType Complex Type

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

```
<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice
    maxOccurs="unbounded"
    minOccurs="1"
  >
    <xs:element name="EventType"
      type="t:NotificationEventTypeType"
    />
  </xs:choice>
</xs:complexType>
```

#### Child Elements

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

### 3.1.4.3.2.5 t:PullSubscriptionRequestType Complex Type

The [PullSubscriptionRequestType](#) complex type specifies a subscription to a pull-based event notification subscription.

```
<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>
```

#### Child Elements

Element	Type	Description
Timeout	<a href="#">t:SubscriptionTimeoutType</a>	Specifies the duration that a subscription can remain idle without a GetEvents request from the client.

### 3.1.4.3.2.6 t:PushSubscriptionRequestType Complex Type

The [PushSubscriptionRequestType](#) complex type specifies a subscription to a push-based event notification.

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

Child Elements

Element	Type	Description
StatusFrequency	<a href="#">t:SubscriptionStatusFrequencyType</a>	Specifies the frequency, specified in minutes, in which push notification status will be sent to the client when the subscription is idle.
URL	xs:string	Specifies the location of the client Web service for push notifications.

### 3.1.4.3.3 Elements

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

#### 3.1.4.3.3.1 Subscribe Element

The [Subscribe](#) element specifies the properties that are used to create subscriptions.

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

### 3.1.4.3.3.2 SubscribeResponse Element

The [tns:SubscribeResponse](#) element specifies a response to a [Subscribe](#) operation (section [3.1.4.3](#)) request.

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

### 3.1.4.3.4 Messages

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

#### 3.1.4.3.4.1 tns:SubscribeSoapIn Message

The SubscribeSoapIn message contains four parts, as described in the following table.

Part Name	Element/Type	Description
request	<a href="#">tns:Subscribe</a>	Specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	Specifies the user whom the client application is impersonating.
MailboxCulture	<a href="#">t:MailboxCulture</a>	Specifies the culture to use for accessing the mailbox. The cultures are defined by <a href="#">[RFC3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	Specifies the schema version for the <b>Subscribe</b> operation (section <a href="#">3.1.4.3</a> ) request.

#### 3.1.4.3.4.2 tns:SubscribeSoapOut Message

The SubscribeSoapOut message contains two parts, as described in the following table.

Part Name	Element/Type	Description
SubscribeResult	<a href="#">tns:SubscribeResponse</a>	Specifies the response message.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	Specifies the schema version for the <b>Subscribe</b> operation (section <a href="#">3.1.4.3</a> ) message.

### 3.1.4.4 Unsubscribe

The [Unsubscribe](#) operation removes a subscription to notifications of events on mailboxes.

Request

Message Format	Description
<a href="#">tns:UnsubscribeSoapIn</a>	Specifies the SOAP message that requests the subscription removal.

Response

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

### 3.1.4.4.1 Complex Types

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

#### 3.1.4.4.1.1 m:UnsubscribeResponseType Complex Type

The [UnsubscribeResponseType](#) complex type specifies a response to an [Unsubscribe](#) operation (section [3.1.4.4](#)) request.

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

#### 3.1.4.4.1.2 m:UnsubscribeType Complex Type

The [UnsubscribeType](#) complex type specifies a request to end a pull notification subscription.

```
<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>
```

Child Elements

Element	Type	Description
SubscriptionId	<a href="#">t:SubscriptionIdType</a>	Specifies the subscription identifier of a subscription to cancel.

### 3.1.4.4.2 Elements

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



### 3.1.4.4.2.1 Unsubscribe Element

The [Unsubscribe](#) element specifies the properties used to unsubscribe from a subscription.

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

### 3.1.4.4.2.2 UnsubscribeResponse Element

The [UnsubscribeResponse](#) element specifies a response to an [Unsubscribe](#) operation (section [3.1.4.4](#)) request.

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

### 3.1.4.4.3 Messages

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

#### 3.1.4.4.3.1 tns:UnsubscribeSoapIn Message

The UnsubscribeSoapIn message contains four parts, as described in the following table.

Part Name	Element/Type	Description
request	<a href="#">tns:Unsubscribe</a>	Specifies the request.
Impersonation	<a href="#">t:ExchangeImpersonation</a>	Specifies the user whom the client application is impersonating.
MailboxCulture	<a href="#">t:MailboxCulture</a>	Specifies the culture to use for accessing the mailbox. The cultures are defined by <a href="#">[RFC3066]</a> .
RequestVersion	<a href="#">t:RequestServerVersion</a>	Specifies the schema version for the <b>Unsubscribe</b> operation (section <a href="#">3.1.4.4</a> ) request.

#### 3.1.4.4.3.2 tns:UnsubscribeSoapOut Message

The UnsubscribeSoapOut message contains two parts, as described in the following table.

Part Name	Element/Type	Description
UnsubscribeResult	<a href="#">tns:UnsubscribeResponse</a>	Specifies the response message.
ServerVersion	<a href="#">t:ServerVersionInfo</a>	Specifies the schema version for the UnsubscribeSoapOut message.

### **3.1.5 Timer Events**

None.

### **3.1.6 Other Local Events**

None.

## **3.2 Client Details**

None.

### **3.2.1 Abstract Data Model**

None.

### **3.2.2 Timers**

None.

### **3.2.3 Initialization**

None.

### **3.2.4 Message Processing Events and Sequencing**

None.

### **3.2.5 Timer Events**

None.

### **3.2.6 Other Local Events**

None.

## 4 Protocol Examples

None.

PRELIMINARY

## 5 Security

### 5.1 Security Considerations for Implementers

The Notifications Web Service protocol does not use any additional security mechanisms.

### 5.2 Index of Security Parameters

None.

PRELIMINARY

## 6 Appendix A: Full WSDL

The following table lists the **XML** files that are required in order to implement the functionality that is specified in this document. The contents of each file are included in this section.

File name	Description	Section
MS-OXWSNTIF.wsdl	Contains the <b>WSDL</b> for the implementation of this protocol.	<a href="#">6.1</a>
MS-OXWSNTIF-types.xsd	Contains the XML schema type definitions that are used in this protocol.	<a href="#">6.2</a>
MS-OXWSNTIF-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	<a href="#">6.3</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.

### 6.1 WSDL

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="Exchange2010_SP1"
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"/>
      <!-- Add global elements and types from messages.xsd -->
    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2010_SP1"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
      <!-- Add global elements and types from types.xsd -->
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ExchangeServicePortType">
    <wsdl:operation name="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: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: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:definitions>

```

## 6.2 Types Schema

This section contains 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 6.3

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010_SP1" 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: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: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>

```

### 6.3 Messages Schema

This section contains the contents of the MS-OXWSNTIF-messages.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 schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-messages.xsd	<a href="#">[MS-OXWSCDATA]</a> section 6.2

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  elementFormDefault="qualified" version="Exchange2010_SP1" id="messages">
  <!-- CHANGE THE SCHEMA LOCATION TO REFLECT THE TYPES XSD ASSOCIATED WITH THIS DOCUMENT
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
  schemaLocation="types.xsd"/>
  <xs:include/>
  -->
  <xs: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>
<xs:complexType name="GetEventsResponseType">
    <xs:complexContent>
        <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
</xs:complexType>
<xs:element name="GetEventsResponse" type="m:GetEventsResponseType"/>
<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"/>
<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 Appendix B: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 2010 SP1 Beta

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

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

PRELIMINARY

## 8 Change Tracking

This section identifies changes made to [MS-OXWSNTIF] protocol documentation between February 2010 and May 2010 releases. Changes are classed as major, minor, or editorial.

**Major** changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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

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

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

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

- New content added.
- Content update.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

**Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

**Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
<a href="#">1.3 Overview</a>	Updated the section title.	N	Content updated for template compliance.
<a href="#">2.2.3.2 m:GetStreamingEventsResponseMessageType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">2.2.3.4 t:BaseNotificationEventType Complex Type</a>	Added minOccurs value.	N	New content added.
<a href="#">2.2.3.8 t:NotificationType Complex Type</a>	Added the FreeBusyChangedEvent element.	Y	New content added.
<a href="#">3.1 ExchangeServicePortType Server Details</a>	Added the GetStreamingEvents operation.	N	Content update.
<a href="#">3.1.4 Message Processing Events and Sequencing</a>	Added the GetStreamingEvents operation.	N	Content update.
<a href="#">3.1.4.2 GetStreamingEvents</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.1 Simple Types</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.1.1 t:StreamingSubscriptionConnectionTimeoutType Simple Type</a>	Added new section.	Y	New content added.



Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
<a href="#">3.1.4.2.1.2</a> <a href="#">t:ConnectionStatusType Simple Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2</a> <a href="#">Complex Types</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2.1</a> <a href="#">m:GetStreamingEventsType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2.2</a> <a href="#">m:GetStreamingEventsResponseType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2.3</a> <a href="#">t:StreamingSubscriptionRequestType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2.4</a> <a href="#">t:NonEmptyArrayOfSubscriptionIdsType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.2.5</a> <a href="#">t:NonEmptyArrayOfNotificationsType Complex Type</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.3</a> <a href="#">Elements</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.3.1</a> <a href="#">GetStreamingEvents Element</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.3.2</a> <a href="#">GetStreamingEventsResponse Element</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.4</a> <a href="#">Messages</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.4.1</a> <a href="#">tns:GetStreamingEventsSoapIn</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.2.4.2</a> <a href="#">tns:GetStreamingEventsSoapOut</a>	Added new section.	Y	New content added.
<a href="#">3.1.4.3.1.1</a> <a href="#">t:NotificationEventTypeType Simple Type</a>	Added the FreeBusyChangedEvent enumeration.	Y	New content added.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
<a href="#">3.1.4.3.2.2 m:SubscribeType Complex Type</a>	Added the StreamingSubscriptionRequest element.	Y	New content added.
<a href="#">6.1 WSDL</a>	Updated to latest WSDL.	Y	Content update.
<a href="#">6.2 Types Schema</a>	Updated to latest types schema.	Y	Content update.
<a href="#">6.3 Messages Schema</a>	Updated to latest messages schema.	Y	Content update.
<a href="#">8 Change Tracking</a>		Y	New content added.

## 9 Index

### C

[Change tracking](#) 47

### I

[Introduction](#) 6

### O

[Overview](#) 7

### T

[Tracking changes](#) 47

PRELIMINARY