[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.

- **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.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	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.
08/04/2010	2.1	Minor	Clarified the meaning of the technical content.
11/03/2010	3.0	Major	Significantly changed the technical content.
03/18/2011	3.1	Minor	Clarified the meaning of the technical content.
08/05/2011	4.0	Major	Significantly changed the technical content.
10/07/2011	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	5.0	Major	Significantly changed the technical content.
04/27/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	5.1	Minor	Clarified the meaning of the technical content.
10/08/2012	6.0	Major	Significantly changed the technical content.
02/11/2013	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/26/2013	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.

Table of Contents

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 RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

The following terms are defined in [MS-GLOS]:

> **Hypertext Transfer Protocol (HTTP)**
> **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**
> **SOAP**
> **SOAP action**
> **SOAP body**
> **SOAP header**
> **XML namespace**

The following terms are defined in [MS-OXGLOS]:

> **endpoint**
> **free/busy status**
> **mailbox**
> **Uniform Resource Locator (URL)**
> **Web Services Description Language (WSDL)**
> **WSDL message**
> **WSDL port type**
> **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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1   Normative References

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

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

[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

[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 Language", 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.S., Beech, D., Maloney, M., Eds., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

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

1.2.2   Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

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

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary".

[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".

[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

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-OXDSCLI], to identify the target **endpoint (4)** 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.
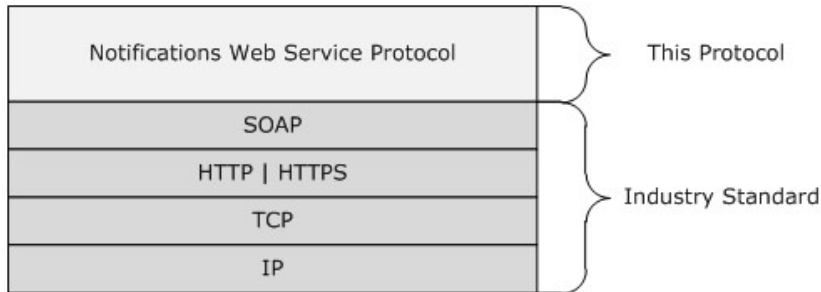
Figure 1: This protocol in relation to other protocols

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 (4) **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 (4) 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 2.2.3.9, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [MS-OXWSCDATA] section 2.2.3.10.

- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.

- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are 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. 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	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	
m	http://schemas.microsoft.com/exchange/services/2006/messages	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

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
BaseNotificationEventType	Specifies the base type for event notifications.
BaseObjectChangedEventType	Specifies identification and time stamp details for create, delete, and new mail events.
GetEventsResponseMessageType	Specifies the status and result of a single **GetEvents** request.
GetStreamingEventsResponseMessageType	Specifies the status and result of a single **GetStreamingEvents** operation request.
ModifiedEventType	Specifies an event in which an item or folder is modified.
MovedCopiedEventType	Specifies a moved or copied event notification.
NotificationType	Specifies information about the subscription and the events that have occurred since the last notification.
SubscribeResponseMessageType	Specifies the status and result of a single **Subscribe** 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 2.2.4.57.

```
<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
Notification	**t:NotificationType** (section 2.2.4.8)	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 2.2.4.57.

```
<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
Notifications	**NonEmptyArrayOfNotificationsType** (section 3.1.4.2.3.5)	Specifies a list of information about the subscription and the events that have occurred since the last notification.
ErrorSubscriptionIds	**NonEmptyArrayOfSubscriptionIdsType** (section 3.1.4.2.3.4)	Specifies a list of subscription IDs that are invalid.
ConnectionStatus	**ConnectionStatusType** (section 3.1.4.2.4.2)	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 2.2.4.57.

```xml
<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
SubscriptionId	**t:SubscriptionIdType** (section 2.2.5.2)	Specifies the identifier for a subscription.
Watermark	**t:WatermarkType** (section 2.2.5.1)	Specifies an event bookmark in the **mailbox** event queue.

2.2.4.4   t:BaseNotificationEventType Complex Type

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

```xml
<xs:complexType name="BaseNotificationEventType"
  minOccurs="0"
>
  <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
Watermark	t:WatermarkType (section 2.2.5.1)	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
TimeStamp	**xs:dateTime** [XMLSCHEMA2]	Specifies the time stamp of a changed item or folder mailbox event.
FolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.31)	Specifies a **FolderIdType** object that identifies the changed store item.
ItemId	**t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.17)	Specifies an **ItemIdType** object that identifies the changed store item.
ParentFolderId	**t:FolderIdType**	Specifies the parent folder identifier of the

Element name	Type	Description
		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>
```

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

Element name	Type	Description
UnreadCount	**xs:int** [XMLSCHEMA2]	Specifies the count of unread items in a particular folder.<3>

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
OldFolderId	**t:FolderIdType** ([MS-OXWSCDATA] section 2.2.4.31)	Specifies the old identifier for a copied or moved folder.
OldItemId	**t:ItemIdType** ([MS-OXWSCORE] section 2.2.4.17)	Specifies the old identifier for a copied or moved item.
OldParentFolderId	**t:FolderIdType**	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"
       />
```

Release: November 18, 2013

```
            <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
SubscriptionId	**t:SubscriptionIdType** (section 2.2.5.2)	Specifies the identifier for a subscription.
PreviousWatermark	**t:WatermarkType** (section 2.2.5.1)	Specifies the watermark of the latest event that was successfully communicated to the client for the subscription.<4>
MoreEvents	**xs:boolean** [XMLSCHEMA2]	Specifies a value that indicates whether more events to be delivered to the client currently exist in the queue. <5>
CopiedEvent	**t:MovedCopiedEventType** (section 2.2.4.7)	Specifies a copied event notification.<6>
CreatedEvent	**t:BaseObjectChangedEventType** (section 2.2.4.5)	Specifies a created event notification.
DeletedEvent	**t:BaseObjectChangedEventType**	Specifies a deleted event notification.
ModifiedEvent	**t:ModifiedEventType** (section 2.2.4.6)	Specifies a modified event notification.
MovedEvent	**t:MovedCopiedEventType**	Specifies a moved event notification.
NewMailEvent	**t:BaseObjectChangedEventType**	Specifies a new e-mail event notification.
StatusEvent	**t:BaseNotificationEventType** (section 2.2.4.4)	Specifies a status event notification.
FreeBusyChangedEvent	**t:BaseObjectChangedEventType**	Specifies an event in which an item's **free/busy status** has changed.<7>

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
WatermarkType	Specifies an event bookmark in the mailbox event queue.
SubscriptionIdType	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
GetEvents	Gets events for a pull notification subscription.
GetStreamingEvents	Gets events for a streaming notification subscription.
Subscribe	Creates a subscription to notifications of events on mailboxes.
Unsubscribe	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
GetEventsSoapIn	Specifies the **GetEvents** operation WSDL message.
GetEventsSoapOut	Specifies the **GetEvents** 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
request	**tns:GetEvents** (section 3.1.4.1.2.1)	Specifies the **SOAP body** of the request.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a **SOAP header** that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.6)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.9)	Specifies a SOAP header that identifies the schema version for the **GetEvents** operation (section 3.1.4.1) 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
GetEventsResult	**tns:GetEventsResponse** (section 3.1.4.1.2.2)	Specifies the SOAP body of the response message.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.10)	Specifies a SOAP header that identifies the schema version for the **GetEventsSoapOut** 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
GetEvents	Specifies a request to the **GetEvents** operation that is used by pull clients to request notifications from the server.
GetEventsResponse	Specifies a response to the **GetEvents** 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
GetEventsResponseType	Specifies the response from the **GetEvents** operation.
GetEventsType	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 2.2.4.16.

```
<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 2.2.4.15.

```
<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
SubscriptionId	**t:SubscriptionIdType** (section 2.2.5.2)	Specifies the identifier of a subscription.
Watermark	**t:WatermarkType** (section 2.2.5.1)	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
GetStreamingEventsSoapIn	Specifies the **GetStreamingEvents** operation WSDL message.
GetStreamingEventsSoapOut	Specifies the **GetStreamingEvents** 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
request	**tns:GetStreamingEvents** (section 3.1.4.2.2.1)	Specifies the SOAP body of the request.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.6)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.9)	Specifies a SOAP header that identifies the schema version for the **GetStreamingEvents** 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
GetStreamingEventsResult	**GetStreamingEventsResponse** (section 3.1.4.2.2.2)	Specifies the SOAP body of the response message.

Part name	Element/type	Description
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)	Specifies a SOAP header that identifies the schema version for the **GetStreamingEvents** operation (section 3.1.4.2) 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
GetStreamingEvents	Specifies the operation that is used by clients to request streaming notifications from the server.
GetStreamingEventsResponse	Specifies a response to a **GetStreamingEvents** 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
GetStreamingEventsType	Specifies a request to get streaming notifications from the server.
GetStreamingEventsResponseType	Specifies a response from the **GetStreamingEvents** operation.

Message name	Description
StreamingSubscriptionRequestType	Specifies a subscription to streaming event notifications.
NonEmptyArrayOfSubscriptionIdsType	Specifies an array of invalid subscription IDs.
NonEmptyArrayOfNotificationsType	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 2.2.4.15.

```
<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
SubscriptionIds	**t:NonEmptyArrayOfSubscriptionIdsType** (section 3.1.4.2.3.4)	Specifies the identifier for a subscription that is queried for events.
ConnectionTimeout	**t:StreamingSubscriptionConnectionTimeoutType** (section 3.1.4.2.4.1)	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 2.2.4.16.

```
<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
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** ([MS-OXWSFOLD] 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.
EventTypes	**t:NonEmptyArrayOfNotificationEventTypesType** (section 3.1.4.3.3.4)	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 invalid 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
SubscriptionId	**t:SubscriptionIdType** (section 2.2.5.2)	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"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
Notification	**t:NotificationType** (section 2.2.4.8)	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
StreamingSubscriptionConnectionTimeoutType	Specifies the number of minutes to keep a connection open.
ConnectionStatusType	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
OK	The connection is good.
Closed	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
SubscribeSoapIn	Specifies the **Subscribe** operation WSDL message.
SubscribeSoapOut	Specifies the **Subscribe** 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
request	**tns:Subscribe** (section 3.1.4.3.2.1)	Specifies the SOAP body of the request.
Impersonation	**t:ExchangeImpersonation** ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	**t:MailboxCulture** ([MS-OXWSCDATA] section 2.2.3.6)	Specifies a SOAP header that identifies the culture to use for accessing the mailbox. The cultures are defined in [RFC3066].
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.9)	Specifies a SOAP header that identifies the schema version for the **Subscribe** 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
SubscribeResult	**tns:SubscribeResponse** (section 3.1.4.3.2.2)	Specifies the SOAP body of the response message.
ServerVersion	**t:ServerVersionInfo** ([MS-OXWSCDATA] section 2.2.3.10)	Specifies a SOAP header that identifies the schema version for the **Subscribe** operation (section 3.1.4.3) 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
Subscribe	Specifies the properties that are used to create subscriptions.
SubscribeResponse	Specifies a response to a **Subscribe** 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
SubscribeResponseType	Specifies a response to a **Subscribe** operation.
SubscribeType	Specifies a request to subscribe to notifications of events on mailboxes.
BaseSubscriptionRequestType	Specifies the base class for descendant classes that form the notification requests.
NonEmptyArrayOfNotificationEventTypesType	Specifies an array of event notification types that are used to create a subscription.
PullSubscriptionRequestType	Specifies a subscription to a pull-based event notification subscription.
PushSubscriptionRequestType	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 2.2.4.16.

```
<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 2.2.4.15.

```
<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
PullSubscriptionRequest	**t:PullSubscriptionRequestType** (section 3.1.4.3.3.5)	Specifies a pull subscription type.
PushSubscriptionRequest	**t:PushSubscriptionRequestType** (section 3.1.4.3.3.6)	Specifies a push subscription type.
StreamingSubscriptionRequest	**t:StreamingSubscriptionRequestType** (section 3.1.4.2.3.3)	Specifies a subscription to a streaming event notification.<9>

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

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

Element name	Type	Description
FolderIds	**t:NonEmptyArrayOfBaseFolderIdsType** ([MS-OXWSFOLD] 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.
EventTypes	**t:NonEmptyArrayOfNotificationEventTypesType**	Specifies the monitored event types for a notification

Element name	Type	Description
	(section 3.1.4.3.3.4)	subscription.
Watermark	**t:WatermarkType** (section 2.2.5.1)	Specifies an event bookmark in the mailbox event queue.

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.<10>

```
<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice
    maxOccurs="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
EventType	**t:NotificationEventTypeType** (section 3.1.4.3.4.1)	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
Timeout	**t:SubscriptionTimeoutType** (section 3.1.4.3.4.2)	Specifies the amount of time that a subscription can remain idle without a **GetEvents** 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
StatusFrequency	**t:SubscriptionStatusFrequencyType** (section 3.1.4.3.4.3)	Specifies the interval, specified in minutes, at which push notification status will be sent to the client when the subscription is idle.
URL	**xs:string** [XMLSCHEMA2]	Specifies the location of the client Web service for push notifications.
CallerData	**xs:string**	Specifies information about the caller.<11>

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
NotificationEventTypeType	Specifies the event types that are reported in a notification.

Simple type name	Description
SubscriptionTimeoutType	Specifies the amount of time that a subscription can remain idle without a **GetEvents** operation request from the client.
SubscriptionStatusFrequencyType	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 **NotificationEventTypeType** simple type.

Value name	Meaning
CopiedEvent	The notification is a copied event.
CreatedEvent	The notification is a created event.
DeletedEvent	The notification is a deleted event.
ModifiedEvent	The notification is a modified event.
MovedEvent	The notification is a moved event.
NewMailEvent	The notification is a new mail event.

Value name	Meaning
FreeBusyChangedEvent	The notification is a free/busy changed event.<12>

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 subscription to notifications of events on mailboxes.

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
UnsubscribeSoapIn	Specifies the **Unsubscribe** operation WSDL message.
UnsubscribeSoapOut	Specifies the **Unsubscribe** 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
request	tns:Unsubscribe (section 3.1.4.4.2.1)	Specifies the SOAP body of the request.
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating.
MailboxCulture	t:MailboxCulture ([MS-OXWSCDATA]	Specifies a SOAP header that identifies the

Part name	Element/type	Description
	section 2.2.3.6)	culture to use for accessing the mailbox. The cultures are defined in [RFC3066].
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.9)	Specifies a SOAP header that identifies the schema version for the **Unsubscribe** 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
UnsubscribeResult	tns:UnsubscribeResponse (section 3.1.4.4.2.2)	Specifies the SOAP body of the response message.
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.10)	Specifies a SOAP header that identifies the schema version for the **UnsubscribeSoapOut** 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
Unsubscribe	Specifies the properties used to end a subscription.
UnsubscribeResponse	Specifies a response to an **Unsubscribe** 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"
 />
```

3.1.4.4.2.2 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
UnsubscribeResponseType	Specifies a response to an **Unsubscribe** operation request.
UnsubscribeType	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 2.2.4.16.

```
<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 pull notification subscription. The **UnsubscribeType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.15.

```
<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>
```

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

Element name	Type	Description
SubscriptionId	**t:SubscriptionIdType** (section 2.2.5.2)	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.

Release: November 18, 2013

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.	7.1
MS-OXWSNTIF-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

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="Exchange2013"
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="Exchange2013"
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: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>
```

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:	7.1
Types schema	t:	7.2

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	[MS-OXWSCDATA] section 7.1

```xml
<?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="Exchange2013" 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: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	[MS-OXWSCDATA] 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="Exchange2013" 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>
```

```xml
            <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: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

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: The **GetStreamingEventsResponseMessageType** complex type is not part of the schema for Exchange 2010 or Exchange 2013.

<2> Section 2.2.4.4: Exchange 2010 and Exchange 2013 do not include the **minOccurs** value.

<3> Section 2.2.4.6: In Exchange 2010 and Exchange 2013, the **UnreadCount** element includes a **maxOccurs** value of 1.

<4> Section 2.2.4.8: Exchange 2010 and Exchange 2013 do not include the minOccurs value for the element.

<5> Section 2.2.4.8: Exchange 2010 and Exchange 2013 do not include the minOccurs value for this element.

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

<7> Section 2.2.4.8: Exchange 2010 and Exchange 2013 do not include the **FreeBusyChangedEvent** element.

<8> Section 3.1.4.2: Exchange 2010 and Exchange 2013 do not include the **GetStreamingEvents** operation.

<9> Section 3.1.4.3.3.2: Exchange 2010 and Exchange 2013 do not include the **StreamingSubscriptionRequest** element.

<10> Section 3.1.4.3.3.4: In Exchange 2010 and Exchange 2013, the **xs:choice** element has a **minOccurs** value of 1.

<11> Section 3.1.4.3.3.6: Exchange 2007 and Exchange 2010 do not support the **CallerData** element.

<12> Section 3.1.4.3.4.1: Exchange 2010 and Exchange 2013 do not include the **FreeBusyChangedEvent** enumeration value.

9   Change Tracking

This section identifies changes that were made to the [MS-OXWSNTIF] protocol document between the July 2013 and November 2013 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

▪ Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

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

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

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

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.1 Namespaces	Added entries to the namespaces table for the namespaces represented by the "m" and "xs" prefixes.	N	Content updated.
2.2.4 Complex Types	Added details for the GetStreamingEventsResponseMessageType, BaseNotificationEventType, and BaseObjectChangedEventType complex types to the table.	N	New content added.
3.1.4.2.1.1 tns:GetStreamingEventsSoapIn	Updated the SOAP action value from http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents to http://schemas.microsoft.com/exchange/services/2006/messages/GetStreamingEvents.	N	Content updated.
3.1.4.2.1.2 tns:GetStreamingEventsSoapOut	Updated the SOAP action value from http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents to http://schemas.microsoft.com/exchange/services/2006/messages/GetStreamingEvents.	N	Content updated.

10 Index