

[MS-OXWSNTIF]: Notifications Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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 |

Table of Contents

| | |
|---|-----------|
| 1 Introduction | 5 |
| 1.1 Glossary | 5 |
| 1.2 References | 5 |
| 1.2.1 Normative References | 5 |
| 1.2.2 Informative References | 6 |
| 1.3 Protocol Overview | 6 |
| 1.4 Relationship to Other Protocols | 6 |
| 1.5 Prerequisites/Preconditions | 7 |
| 1.6 Applicability Statement | 7 |
| 1.7 Versioning and Capability Negotiation | 7 |
| 1.8 Vendor Extensible Fields | 7 |
| 1.9 Standards Assignments | 7 |
| 2 Messages | 8 |
| 2.1 Transport | 8 |
| 2.2 Common Message Syntax | 8 |
| 2.2.1 Namespaces | 8 |
| 2.2.2 Simple Types | 8 |
| 2.2.2.1 t:WatermarkType Simple Type | 8 |
| 2.2.2.2 t:SubscriptionIdType Simple Type | 9 |
| 2.2.3 Complex Types | 9 |
| 2.2.3.1 m:GetEventsResponseMessageType Complex Type | 9 |
| 2.2.3.2 m:SubscribeResponseMessageType Complex Type | 10 |
| 2.2.3.3 t:BaseNotificationEventType Complex Type | 10 |
| 2.2.3.4 t:BaseObjectChangedEventType Complex Type | 11 |
| 2.2.3.5 t:ModifiedEventType Complex Type | 11 |
| 2.2.3.6 t:MovedCopiedEventType Complex Type | 12 |
| 2.2.3.7 t:NotificationType Complex Type | 13 |
| 2.2.4 Elements | 14 |
| 2.2.5 Attributes | 14 |
| 2.2.6 Groups | 14 |
| 2.2.7 Attribute Groups | 14 |
| 2.2.8 Messages | 14 |
| 3 Protocol Details | 15 |
| 3.1 ExchangeServicePortType Server Details | 15 |
| 3.1.1 Abstract Data Model | 15 |
| 3.1.2 Timers | 15 |
| 3.1.3 Initialization | 15 |
| 3.1.4 Message Processing Events and Sequencing | 15 |
| 3.1.4.1 GetEvents | 15 |
| 3.1.4.1.1 Complex Types | 16 |
| 3.1.4.1.1.1 m:GetEventsResponseType Complex Type | 16 |
| 3.1.4.1.1.2 m:GetEventsType Complex Type | 16 |
| 3.1.4.1.2 Elements | 17 |
| 3.1.4.1.2.1 GetEvents Element | 17 |
| 3.1.4.1.2.2 GetEventsResponse Element | 17 |
| 3.1.4.1.3 Messages | 17 |
| 3.1.4.1.3.1 tns:GetEventsSoapIn Message | 17 |
| 3.1.4.1.3.2 tns:GetEventsSoapOut Message | 18 |

| | | |
|-------------|---|-----------|
| 3.1.4.2 | Subscribe..... | 18 |
| 3.1.4.2.1 | Simple Types | 18 |
| 3.1.4.2.1.1 | t:NotificationEventTypeType Simple Type | 18 |
| 3.1.4.2.1.2 | t:SubscriptionTimeoutType Simple Type | 19 |
| 3.1.4.2.1.3 | t:SubscriptionStatusFrequencyType Simple Type | 19 |
| 3.1.4.2.2 | Complex Types..... | 20 |
| 3.1.4.2.2.1 | m:SubscribeResponseType Complex Type | 20 |
| 3.1.4.2.2.2 | m:SubscribeType Complex Type..... | 20 |
| 3.1.4.2.2.3 | t:BaseSubscriptionRequestType Complex Type | 21 |
| 3.1.4.2.2.4 | t:NonEmptyArrayOfNotificationEventTypesType Complex Type..... | 21 |
| 3.1.4.2.2.5 | t:PullSubscriptionRequestType Complex Type | 22 |
| 3.1.4.2.2.6 | t:PushSubscriptionRequestType Complex Type | 22 |
| 3.1.4.2.3 | Elements..... | 23 |
| 3.1.4.2.3.1 | Subscribe Element | 23 |
| 3.1.4.2.3.2 | SubscribeResponse Element..... | 23 |
| 3.1.4.2.4 | Messages | 23 |
| 3.1.4.2.4.1 | tns:SubscribeSoapIn Message | 23 |
| 3.1.4.2.4.2 | tns:SubscribeSoapOut Message..... | 24 |
| 3.1.4.3 | Unsubscribe | 24 |
| 3.1.4.3.1 | Complex Types..... | 24 |
| 3.1.4.3.1.1 | m:UnsubscribeResponseType Complex Type..... | 24 |
| 3.1.4.3.1.2 | m:UnsubscribeType Complex Type | 25 |
| 3.1.4.3.2 | Elements..... | 25 |
| 3.1.4.3.2.1 | Unsubscribe Element..... | 25 |
| 3.1.4.3.2.2 | UnsubscribeResponse Element | 25 |
| 3.1.4.3.3 | Messages | 26 |
| 3.1.4.3.3.1 | tns:UnsubscribeSoapIn Message..... | 26 |
| 3.1.4.3.3.2 | tns:UnsubscribeSoapOut Message | 26 |
| 3.1.5 | Timer Events..... | 26 |
| 3.1.6 | Other Local Events | 26 |
| 3.2 | Client Details..... | 26 |
| 3.2.1 | Abstract Data Model..... | 26 |
| 3.2.2 | Timers | 26 |
| 3.2.3 | Initialization | 26 |
| 3.2.4 | Message Processing Events and Sequencing | 27 |
| 3.2.5 | Timer Events..... | 27 |
| 3.2.6 | Other Local Events | 27 |
| 4 | Protocol Examples | 28 |
| 5 | Security..... | 29 |
| 5.1 | Security Considerations for Implementers..... | 29 |
| 5.2 | Index of Security Parameters | 29 |
| 6 | Appendix A: Full WSDL..... | 30 |
| 6.1 | WSDL..... | 30 |
| 6.2 | Types Schema..... | 32 |
| 6.3 | Messages Schema | 35 |
| 7 | Appendix B: Product Behavior | 38 |
| 8 | Change Tracking | 39 |
| 9 | Index..... | 40 |

1 Introduction

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

1.1 Glossary

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

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

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

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

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

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

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

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

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

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

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

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

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

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, <http://www.w3.org/TR/REC-xml-names/>.

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

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

1.2.2 Informative References

None.

1.3 Protocol Overview

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

1.4 Relationship to Other Protocols

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

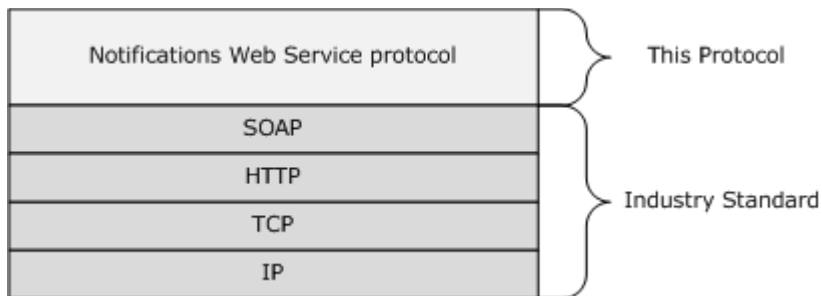


Figure 1: Notification Web Service protocol HTTP stack

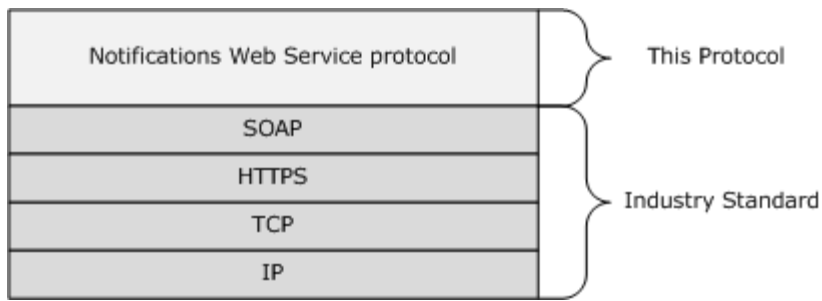


Figure 2: Notification Web Service protocol HTTPS stack

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

Supported Transports: This protocol uses SOAP 1.1, as specified in section [2.1](#).

Protocol Versions: This protocol specifies only one **WSDL port type** version.

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

Localization: This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).

Capability Negotiation: None.

1.8 Vendor Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

| Prefix | Namespace URI | Reference |
|-----------------|---|------------------------------|
| soap | http://schemas.xmlsoap.org/wsdl/soap/ | [SOAP1.1] |
| tns | http://schemas.microsoft.com/exchange/services/2006/messages | [MS-OXWSNTIF] |
| s | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1] |
| targetNamespace | http://schemas.microsoft.com/exchange/services/2006/messages | [MS-OXWSNTIF] |
| wsdl | http://schemas.xmlsoap.org/wsdl/ | [WSDL] |
| t | http://schemas.microsoft.com/exchange/services/2006/types | [MS-OXWSNTIF] |

2.2.2 Simple Types

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

| Simple Type | Description |
|--------------------------------------|--|
| t:WatermarkType | Specifies an event bookmark in the mailbox event queue. |
| t:SubscriptionIdType | Specifies the identifier for a subscription. |

2.2.2.1 t:WatermarkType Simple Type

The [WatermarkType](#) simple type specifies an event bookmark in the mailbox event queue.

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


2.2.2.2 t:SubscriptionIdType Simple Type

The [SubscriptionIdType](#) simple type specifies the identifier for a subscription.

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

2.2.3 Complex Types

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

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

2.2.3.1 m:GetEventsResponseMessageType Complex Type

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

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

Child Elements

| Element | Type | Description |
|--------------|------------------------------------|---|
| Notification | t:NotificationType | Specifies information about the subscription and the events that have occurred since the last notification. |

2.2.3.2 m:SubscribeResponseMessageType Complex Type

The [SubscribeResponseMessageType](#) complex type specifies the status and result of a single [Subscribe](#) operation (section [3.1.4.2](#)) request.

```
<xs:complexType name="SubscribeResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseType"
    >
      <xs:sequence>
        <xs:element name="SubscriptionId"
          type="t:SubscriptionIdType"
          minOccurs="0"
        />
        <xs:element name="Watermark"
          type="t:WatermarkType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

| Element | Type | Description |
|----------------|--------------------------------------|---|
| SubscriptionId | t:SubscriptionIdType | Specifies the identifier for a subscription. |
| Watermark | t:WatermarkType | Specifies an event bookmark in the mailbox event queue. |

2.2.3.3 t:BaseNotificationEventType Complex Type

The [t:BaseNotificationEventType](#) complex type specifies the base type for event notifications.

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

Child Elements

| Element | Type | Description |
|-----------|---------------------------------|---|
| Watermark | t:WatermarkType | Specifies an event bookmark in the mailbox event queue. |

2.2.3.4 t:BaseObjectChangedEventType Complex Type

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

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

Child Elements

| Element | Type | Description |
|----------------|--------------------------------|---|
| TimeStamp | xs:dateTime | Specifies the time stamp of a changed item or folder mailbox event. |
| FolderId | t:FolderIdType | Specifies a FolderIdType object that identifies the changed store item. |
| ItemId | t:ItemIdType | Specifies an ItemIdType object that identifies the changed store item. |
| ParentFolderId | t:FolderIdType | Specifies the parent folder identifier of the changed item or folder. |

2.2.3.5 t:ModifiedEventType Complex Type

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

```
<xs:complexType name="ModifiedEventType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseObjectChangedEventType"
    >
  </xs:extension>
</xs:complexType>
```

```

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

```

Child Elements

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

2.2.3.6 t:MovedCopiedEventType Complex Type

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

```

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

```

Child Elements

| Element | Type | Description |
|-------------------|--------------------------------|---|
| OldFolderId | t:FolderIdType | Specifies the old identifier for a copied or moved folder. |
| OldItemId | t:ItemIdType | Specifies the old identifier for a copied or moved item. |
| OldParentFolderId | t:FolderIdType | Specifies the identifier of the original parent folder of an item or folder that was moved. |

2.2.3.7 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"
    />
    <xs:element name="MoreEvents"
      type="xs:boolean"
    />
    <xs:choice
      maxOccurs="unbounded"
      minOccurs="1"
    >
      <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:choice>
  </xs:sequence>
</xs:complexType>
```

Child Elements

| Element | Type | Description |
|-------------------|--------------------------------------|--|
| SubscriptionId | t:SubscriptionIdType | Specifies the identifier for a subscription. |
| PreviousWatermark | t:WatermarkType | Specifies the watermark of the latest event that was successfully communicated to the client for the subscription. |
| MoreEvents | xs:boolean | Specifies a value that indicates whether more events to be delivered to the client exist in the |

| Element | Type | Description |
|---------------|--|--|
| | | queue. |
| CopiedEvent | t:MovedCopiedEventType | Specifies a copied event notification. |
| CreatedEvent | t:BaseObjectChangedEventType | Specifies a created event notification. |
| DeletedEvent | t:BaseObjectChangedEventType | Specifies a deleted event notification. |
| ModifiedEvent | t:ModifiedEventType | 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 | Specifies a status event notification. |

2.2.4 Elements

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

2.2.5 Attributes

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

2.2.6 Groups

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

2.2.7 Attribute Groups

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

2.2.8 Messages

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

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

The Notifications Web Service protocol defines a single port type with three operations.

| Operation | Description |
|-----------------------------|---|
| GetEvents | Gets events for a pull notification subscription. |
| Subscribe | Creates a subscription to notifications of events on mailboxes. |
| Unsubscribe | Removes a subscription to notifications of events on mailboxes. |

3.1.1 Abstract Data Model

The Notifications Web Service protocol is a stateless protocol.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing

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

| Operation | Description |
|-----------------------------|---|
| GetEvents | Gets events for a pull 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.

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

Request

| Message Format | Description |
|-------------------------------------|---|
| tns:GetEventsSoapIn | Specifies the SOAP message that requests the events. |

Response

| Message Format | Description |
|--------------------------------------|--|
| tns:GetEventsSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.1.1 Complex Types

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

3.1.4.1.1.1 m:GetEventsResponseType Complex Type

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

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

3.1.4.1.1.2 m:GetEventsType Complex Type

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

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

Child Elements

| Element | Type | Description |
|----------------|--------------------------------------|--|
| SubscriptionId | t:SubscriptionIdType | Specifies the identifier of a subscription. |
| Watermark | t:WatermarkType | Specifies a string that represents an event bookmark in the mailbox event queue. |

3.1.4.1.2 Elements

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

3.1.4.1.2.1 GetEvents Element

The [GetEvents](#) element specifies the operation that is used by pull clients to request notifications from the server.

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

3.1.4.1.2.2 GetEventsResponse Element

The [GetEventsResponse](#) element specifies a response to a [GetEvents](#) operation (section [3.1.4.1](#)) request.

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

3.1.4.1.3 Messages

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

3.1.4.1.3.1 tns:GetEventsSoapIn Message

The [GetEventsSoapIn](#) message contains four parts, as described in the following table.

| Part Name | Element/Type | Description |
|----------------|---|---|
| request | tns:GetEvents | Specifies the request. |
| Impersonation | t:ExchangeImpersonation | Specifies the user whom the client application is impersonating. |
| MailboxCulture | t:MailboxCulture | Specifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066] . |
| RequestVersion | t:RequestServerVersion | Specifies the schema version for the GetEvents operation (section 3.1.4.1) request. |

3.1.4.1.3.2 tns:GetEventsSoapOut Message

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

| Part Name | Element/Type | Description |
|-----------------|---------------------------------------|--|
| GetEventsResult | tns:GetEventsResponse | Specifies the response message. |
| ServerVersion | t:ServerVersionInfo | Specifies the schema version for the GetEventsSoapOut message. |

3.1.4.2 Subscribe

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

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

Request

| Message Format | Description |
|-------------------------------------|--|
| tns:SubscribeSoapIn | Specifies the SOAP message that requests the subscription. |

Response

| Message Format | Description |
|--------------------------------------|--|
| tns:SubscribeSoapOut | Specifies the SOAP message that is returned by the server in response. |

3.1.4.2.1 Simple Types

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

3.1.4.2.1.1 t: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:restriction>
</xs:simpleType>

```

Enumeration

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

| Value | Description |
|---------------|--|
| CopiedEvent | Identifies the notification as a copied event. |
| CreatedEvent | Identifies the notification as a created event. |
| DeletedEvent | Identifies the notification as a deleted event. |
| ModifiedEvent | Identifies the notification as a modified event. |
| MovedEvent | Identifies the notification as a moved event. |
| NewMailEvent | Identifies the notification as a new mail event. |

3.1.4.2.1.2 t:SubscriptionTimeoutType Simple Type

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

```

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

```

3.1.4.2.1.3 t:SubscriptionStatusFrequencyType Simple Type

The [SubscriptionStatusFrequencyType](#) simple type specifies the frequency, in minutes, at which push notification status will be sent to the client when the subscription is idle.

```

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

```

3.1.4.2.2 Complex Types

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

3.1.4.2.2.1 m:SubscribeResponseType Complex Type

The [SubscribeResponseType](#) complex type specifies a response to a [Subscribe](#) operation (section [3.1.4.2](#)) request.

```

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

```

3.1.4.2.2.2 m:SubscribeType Complex Type

The [SubscribeType](#) complex type specifies a request to subscribe to notifications of events on mailboxes.

```

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

```

Child Elements

| Element | Type | Description |
|-------------------------|---|-------------------------------------|
| PullSubscriptionRequest | t:PullSubscriptionRequestType | Specifies a pull subscription type. |
| PushSubscriptionRequest | t:PushSubscriptionRequestType | Specifies a push subscription type. |

3.1.4.2.2.3 t:BaseSubscriptionRequestType Complex Type

The [BaseSubscriptionRequestType](#) complex type specifies the base class for descendant classes that form the notification requests.

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

Child Elements

| Element | Type | Description |
|------------|---|--|
| FolderIds | t:NonEmptyArrayOfBaseFolderIdsType | Specifies an array of folder identifiers that are used to identify folders to monitor for event notifications. |
| EventTypes | t:NonEmptyArrayOfNotificationEventTypesType | Specifies the monitored event types for a notification subscription. |
| Watermark | t:WatermarkType | Specifies an event bookmark in the mailbox event queue. |

3.1.4.2.2.4 t:NonEmptyArrayOfNotificationEventTypesType Complex Type

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

```
<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice
    minOccurs="1"
    maxOccurs="unbounded"
  >
  </xs:choice>
</xs:complexType>
```

```

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

```

Child Elements

| Element | Type | Description |
|-----------|---|--|
| EventType | t:NotificationEventTypeType | Specifies a requested event notification type that is used to create a subscription. |

3.1.4.2.2.5 t:PullSubscriptionRequestType Complex Type

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

```

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

```

Child Elements

| Element | Type | Description |
|---------|---|---|
| Timeout | t:SubscriptionTimeoutType | Specifies the duration that a subscription can remain idle without a GetEvents request from the client. |

3.1.4.2.2.6 t:PushSubscriptionRequestType Complex Type

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

```

<xs:complexType name="PushSubscriptionRequestType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseSubscriptionRequestType"
    >
      <xs:sequence>
        <xs:element name="StatusFrequency"
          type="t:SubscriptionStatusFrequencyType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    />
    <xs:element name="URL"
      type="xs:string"
    />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Child Elements

| Element | Type | Description |
|-----------------|---|--|
| StatusFrequency | t:SubscriptionStatusFrequencyType | Specifies the frequency, specified in minutes, in which push notification status will be sent to the client when the subscription is idle. |
| URL | xs:string | Specifies the location of the client Web service for push notifications. |

3.1.4.2.3 Elements

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

3.1.4.2.3.1 Subscribe Element

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

```

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

```

3.1.4.2.3.2 SubscribeResponse Element

The [tns:SubscribeResponse](#) element specifies a response to a [Subscribe](#) operation (section [3.1.4.2](#)) request.

```

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

```

3.1.4.2.4 Messages

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

3.1.4.2.4.1 tns:SubscribeSoapIn Message

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

| Part Name | Element/Type | Description |
|----------------|---|---|
| request | tns:Subscribe | Specifies the request. |
| Impersonation | t:ExchangeImpersonation | Specifies the user whom the client application is impersonating. |
| MailboxCulture | t:MailboxCulture | Specifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066] . |
| RequestVersion | t:RequestServerVersion | Specifies the schema version for the Subscribe operation (section 3.1.4.2) request. |

3.1.4.2.4.2 tns:SubscribeSoapOut Message

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

| Part Name | Element/Type | Description |
|-----------------|---------------------------------------|---|
| SubscribeResult | tns:SubscribeResponse | Specifies the response message. |
| ServerVersion | t:ServerVersionInfo | Specifies the schema version for the Subscribe operation (section 3.1.4.2) message. |

3.1.4.3 Unsubscribe

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

Request

| Message Format | Description |
|---------------------------------------|--|
| tns:UnsubscribeSoapIn | Specifies the SOAP message that requests the subscription removal. |

Response

| Message Format | Description |
|--|---|
| tns:UnsubscribeSoapOut | Specifies the SOAP message that are returned by the server in response. |

3.1.4.3.1 Complex Types

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

3.1.4.3.1.1 m:UnsubscribeResponseType Complex Type

The [UnsubscribeResponseType](#) complex type specifies a response to an [Unsubscribe](#) operation (section [3.1.4.3](#)) request.

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



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

3.1.4.3.1.2 m:UnsubscribeType Complex Type

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

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

Child Elements

| Element | Type | Description |
|----------------|--------------------------------------|--|
| SubscriptionId | t:SubscriptionIdType | Specifies the subscription identifier of a subscription to cancel. |

3.1.4.3.2 Elements

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

3.1.4.3.2.1 Unsubscribe Element

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

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

3.1.4.3.2.2 UnsubscribeResponse Element

The [UnsubscribeResponse](#) element specifies a response to an [Unsubscribe](#) operation (section [3.1.4.3](#)) request.

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

3.1.4.3.3 Messages

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

3.1.4.3.3.1 tns:UnsubscribeSoapIn Message

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

| Part Name | Element/Type | Description |
|----------------|---|---|
| request | tns:Unsubscribe | Specifies the request. |
| Impersonation | t:ExchangeImpersonation | Specifies the user whom the client application is impersonating. |
| MailboxCulture | t:MailboxCulture | Specifies the culture to use for accessing the mailbox. The cultures are defined by [RFC3066] . |
| RequestVersion | t:RequestServerVersion | Specifies the schema version for the Unsubscribe operation (section 3.1.4.3) request. |

3.1.4.3.3.2 tns:UnsubscribeSoapOut Message

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

| Part Name | Element/Type | Description |
|-------------------|---|--|
| UnsubscribeResult | tns:UnsubscribeResponse | Specifies the response message. |
| ServerVersion | t:ServerVersionInfo | Specifies the schema version for the UnsubscribeSoapOut message. |

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client Details

None.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Message Processing Events and Sequencing

None.

3.2.5 Timer Events

None.

3.2.6 Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

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

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

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

| File name | Description | Section |
|--------------------------|---|---------------------|
| MS-OXWSNTIF.wsdl | Contains the WSDL for the implementation of this protocol. | 6.1 |
| MS-OXWSNTIF-types.xsd | Contains the XML schema type definitions that are used in this protocol. | 6.2 |
| MS-OXWSNTIF-messages.xsd | Contains the XML schema message definitions that are used in this protocol. | 6.3 |

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

6.1 WSDL

This section contains the contents of the MS-OXWSNTIF.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
<wsdl:types>
<xs:schema id="messages" elementFormDefault="qualified" version="Exchange2010"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
<xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
<xs:include schemaLocation="MS-OXWSNTIF-messages.xsd" />
<xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
<!-- Add global elements and types from messages.xsd -->
</xs:schema>
<xs:schema id="types" elementFormDefault="qualified" version="Exchange2010"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
<!-- Add global elements and types from types.xsd -->
</xs:schema>
</wsdl:types>
<wsdl:portType name="ExchangeServicePortType">
<wsdl:operation name="Subscribe">
<wsdl:input message="tns:SubscribeSoapIn" />
<wsdl:output message="tns:SubscribeSoapOut" />
</wsdl:operation>
<wsdl:operation name="Unsubscribe">
```

```

        <wsdl:input message="tns:UnsubscribeSoapIn" />
        <wsdl:output message="tns:UnsubscribeSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="GetEvents">
        <wsdl:input message="tns:GetEventsSoapIn" />
        <wsdl:output message="tns:GetEventsSoapOut" />
    </wsdl:operation>
</wsdl: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: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:definitions>

```

6.2 Types Schema

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

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

| File name | Defining specification |
|------------------------|--|
| MS-OXWSCDATA-types.xsd | [MS-OXWSCDATA] section 6.2 |


```

<?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="Exchange2010" 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"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BaseObjectChangedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseNotificationEventType">
        <xs:sequence>
          <xs:element name="TimeStamp" type="xs:dateTime"/>
          <xs:choice>
            <xs:element name="FolderId" type="t:FolderIdType"/>
            <xs:element name="ItemId" type="t:ItemIdType"/>
          </xs:choice>
          <xs:element name="ParentFolderId" type="t:FolderIdType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="BaseSubscriptionRequestType" abstract="true">
    <xs:sequence>
      <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"
minOccurs="0"/>
      <xs:element name="EventTypes"
type="t:NonEmptyArrayOfNotificationEventTypesType"/>
      <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
  </xs:complexType>
  <xs:complexType name="ModifiedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseObjectChangedEventType">
        <xs:sequence>
          <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MovedCopiedEventType">
    <xs:complexContent>
      <xs:extension base="t:BaseObjectChangedEventType">
        <xs:sequence>
          <xs:choice>
            <xs:element name="OldFolderId" type="t:FolderIdType"/>
            <xs:element name="OldItemId" type="t:ItemIdType"/>
          </xs:choice>
          <xs:element name="OldParentFolderId" type="t:FolderIdType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

```

<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice maxOccurs="unbounded">
    <xs:element name="EventType" type="t:NotificationEventTypeType"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
    <xs:element name="PreviousWatermark" type="t:WatermarkType"/>
    <xs:element name="MoreEvents" type="xs:boolean"/>
    <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: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:restriction>
</xs:simpleType>
<xs:complexType name="PushSubscriptionRequestType">
  <xs:complexContent>
    <xs:extension base="t:BaseSubscriptionRequestType">
      <xs:sequence>
        <xs:element name="StatusFrequency"
type="t:SubscriptionStatusFrequencyType"/>
        <xs:element name="URL" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="PullSubscriptionRequestType">
  <xs:complexContent>
    <xs:extension base="t:BaseSubscriptionRequestType">
      <xs:sequence>
        <xs:element name="Timeout" type="t:SubscriptionTimeoutType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:simpleType name="SubscriptionIdType">
  <xs:restriction base="t:NonEmptyStringType"/>
</xs:simpleType>
<xs:simpleType name="WatermarkType">
  <xs:restriction base="t:NonEmptyStringType"/>
</xs:simpleType>
<xs:simpleType name="SubscriptionTimeoutType">
  <xs:restriction base="xs:int">

```

```

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

```

6.3 Messages Schema

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

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

| File name | Defining specification |
|---------------------------|--|
| MS-OXWSCDATA-messages.xsd | [MS-OXWSCDATA] section 6.3 |

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2010" id="messages">
    <!--      CHANGE THE SCHEMA LOCATION TO REFLECT THE TYPES XSD ASSOCIATED WITH THIS
DOCUMENT
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="types.xsd"/>
    <xs:include/>
-->
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSNTIF-types.xsd"/>
    <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
    <xs:complexType name="GetEventsType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:sequence>
                    <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
                    <xs:element name="Watermark" type="t:WatermarkType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="GetEvents" type="m:GetEventsType"/>
    <xs:complexType name="GetEventsResponseMessageType">
        <xs:complexContent>
            <xs:extension base="m:ResponseMessageType">
                <xs:sequence>

```

```

        <xs:element name="Notification" type="t:NotificationType"
minOccurs="0"/>
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType 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:choice>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<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:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<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="UnsubscribeResponseMessageType">
    <xs:complexContent>
        <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
</xs:complexType>
<xs:element name="UnsubscribeResponse" type="m:UnsubscribeResponseMessageType"/>
</xs:schema>

```


7 Appendix B: Product Behavior

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

- Microsoft Exchange Server 2010

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

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

8 Change Tracking

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

9 Index

C

[Change tracking](#) 39

I

[Introduction](#) 5

O

[Overview](#) 6

T

[Tracking changes](#) 39