[MS-OXWSNTIF]:
Notifications Web Service Protocol Specification

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.

Contents

Release: Thursday, July 29, 2010

Release: Thursday, July 29, 2010

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)
Hyptertext 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-OXWSCDATA] Microsoft Corporation, "Common Web Service Data Types", July 2009.

[MS-OXWSPSNTIF] Microsoft Corporation, "Push Notifications Web Service Protocol Specification", 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

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

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

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

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

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

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

1.2.2 Informative References

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

1.3 Overview

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

1.4 Relationship to Other Protocols

The Notifications Web Service protocol uses SOAP over **HTTP** and SOAP over **HTTPS**, as shown in the following figures.
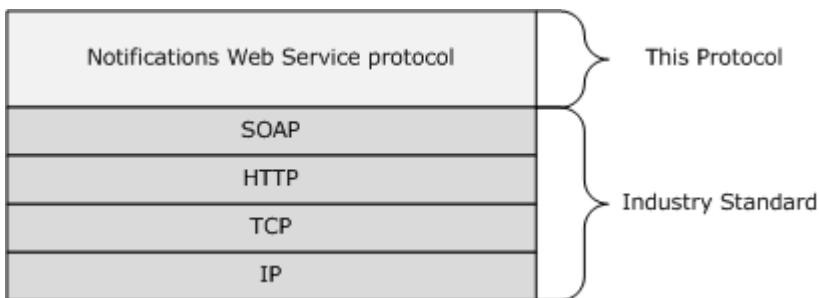
Figure 1: Notification Web Service protocol HTTP stack
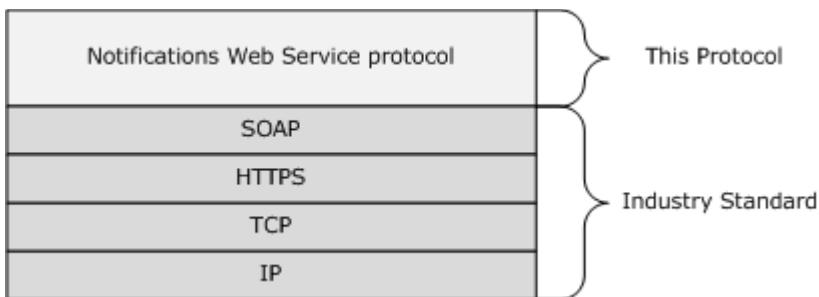
Figure 2: Notification Web Service protocol HTTPS stack

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as specified in section 2.1.

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

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

- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section 3.1.4.

- **Capability Negotiation:** None.

1.8 Vendor Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The SOAP version supported is SOAP 1.1. For details, see [SOAP1.1].

2.2 Common Message Syntax

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
soap	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	

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: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 2.2.3.52.

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

Child Elements

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

2.2.3.3   m:SubscribeResponseMessageType Complex Type

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

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

Child Elements

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

2.2.3.4   t:BaseNotificationEventType Complex Type

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

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

Child Elements

Element	Type	Description
Watermark	t:WatermarkType	Specifies an event bookmark in the mailbox event queue.<2>

2.2.3.5   t:BaseObjectChangedEventType Complex Type

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

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

Child Elements

Element	Type	Description
TimeStamp	xs:dateTime	Specifies the time stamp of a changed item or folder mailbox event.
FolderId	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.

2.2.3.6 t:ModifiedEventType Complex Type

The **ModifiedEventType** complex type specifies an event in which an item or folder is modified.

```
<xs:complexType name="ModifiedEventType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseObjectChangedEventType"
    >
      <xs:sequence>
        <xs:element name="UnreadCount"
          type="xs:int"
          minOccurs="0"
        />
      </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.<3>

2.2.3.7   t:MovedCopiedEventType Complex Type

The MovedCopiedEventType complex type specifies a moved or copied event notification.

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

Child Elements

Element	Type	Description
OldFolderId	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.

2.2.3.8   t:NotificationType Complex Type

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

```
<xs:complexType name="NotificationType">
  <xs:sequence>
    <xs:element name="SubscriptionId"
```

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

Child Elements

Element	Type	Description
SubscriptionId	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.<4>
MoreEvents	xs:boolean	Specifies a value that indicates whether more events to be delivered to the client exist in the queue. <5>
CopiedEvent	t:MovedCopiedEventType	Specifies a copied event notification.<6>

Element	Type	Description
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.
FreeBusyChangedEvent	t:BaseObjectChangedEventType	Specifies an event in which an item's free/busy time has changed.<7>

2.2.4 Elements

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

2.2.5 Attributes

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

2.2.6 Groups

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

2.2.7 Attribute Groups

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

2.2.8 Messages

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

3   Protocol Details

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

3.1   ExchangeServicePortType Server Details

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

Operation	Description
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.

3.1.1   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 (section 3.1.4.1)	Gets events for a pull notification subscription.
GetStreamingEvents (section 3.1.4.2)	Gets events for a streaming notification subscription.
Subscribe (section 3.1.4.3)	Creates a subscription to notifications of events on mailboxes.
Unsubscribe (section 3.1.4.4)	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].

Part Name	Element/Type	Description
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 GetStreamingEvents

The **GetStreamingEvents** operation gets events for a streaming notification subscription.<8>

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

Request

Message Format	Description
tns:GetStreamingEventsSoapIn	Specifies the SOAP message that requests the events.

Response

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

3.1.4.2.1 Simple Types

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

3.1.4.2.1.1 t:StreamingSubscriptionConnectionTimeoutType Simple Type

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

```
<xs:simpleType name="StreamingSubscriptionConnectionTimeoutType">
  <xs:restriction
    base="xs:int"
  >
    <xs:minInclusive
```

```
      value="1"
    />
   <xs:maxInclusive
     value="30"
    />
  </xs:restriction>
 </xs:simpleType>
```

3.1.4.2.1.2   t:ConnectionStatusType Simple Type

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

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

Enumeration

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

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

3.1.4.2.2   Complex Types

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

3.1.4.2.2.1   m:GetStreamingEventsType Complex Type

The **GetStreamingEventsType** complex type specifies a request to get streaming notifications from the server. The **GetStreamingEventsType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.3.14.

```
 <xs:complexType>
   <xs:complexContent>
     <xs:extension
       base="m:BaseRequestType"
     >
       <xs:sequence>
```

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

Child Elements

Element	Type	Description
SubscriptionIds	t:NonEmptyArrayOfSubscriptionIdsType	Specifies the identifier for a subscription that is queried for events.
ConnectionTimeout	t:StreamingSubscriptionConnectionTimeoutType	Specifies the number of minutes to keep a connection open.

3.1.4.2.2.2   m:GetStreamingEventsResponseType Complex Type

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

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

3.1.4.2.2.3   t:StreamingSubscriptionRequestType Complex Type

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

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

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

Child Elements

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

3.1.4.2.2.4   t:NonEmptyArrayOfSubscriptionIdsType Complex Type

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

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

Child Elements

Element	Type	Description
SubscriptionId	t:SubscriptionIdType	Specifies the identifier for a subscription.

3.1.4.2.2.5   t:NonEmptyArrayOfNotificationsType Complex Type

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

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="Notification"
      type="t:NotificationType"
      maxOccurs="unbounded"
      minOccurs="0"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

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

3.1.4.2.3 Elements

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

3.1.4.2.3.1 GetStreamingEvents Element

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

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

3.1.4.2.3.2 GetStreamingEventsResponse Element

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

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

3.1.4.2.4 Messages

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

3.1.4.2.4.1 tns:GetStreamingEventsSoapIn

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

Part name	Element/type	Description
request	tns:GetStreamingEvents	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 3.1.4.3) request.

3.1.4.2.4.2   tns:GetStreamingEventsSoapOut

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

Part name	Element/type	Description
GetStreamingEventsResult	\<GetStreamingEventsResponse\> (section 3.1.4.2.3.2)	Specifies the response message.
ServerVersion	\<ServerVersionInfo\> (section 2.2.4.8)	Specifies the schema version for the **GetStreamingEvents** operation (section 3.1.4.2) message.

3.1.4.3   Subscribe

The Subscribe operation creates a subscription to notifications of events on mailboxes in the mail server database.

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

Request

Message Format	Description
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.

3.1.4.3.1   Simple Types

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

3.1.4.3.1.1   t:NotificationEventTypeType Simple Type

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

```
<xs:simpleType name="NotificationEventTypeType">
  <xs:restriction>
    <xs:enumeration
     value="CopiedEvent"
     />
    <xs:enumeration
     value="CreatedEvent"
     />
```

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

Enumeration

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

Value	Description
CopiedEvent	Identifies the notification as a copied event.
CreatedEvent	Identifies the notification as a created event.
DeletedEvent	Identifies the notification as a deleted event.
ModifiedEvent	Identifies the notification as a modified event.
MovedEvent	Identifies the notification as a moved event.
NewMailEvent	Identifies the notification as a new mail event.
FreeBusyChangedEvent	Identifies the notification as a free/busy changed event.<9>

3.1.4.3.1.2   t:SubscriptionTimeoutType Simple Type

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

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

3.1.4.3.1.3   t:SubscriptionStatusFrequencyType Simple Type

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

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

3.1.4.3.2   Complex Types

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

3.1.4.3.2.1   m:SubscribeResponseType Complex Type

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

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

3.1.4.3.2.2   m:SubscribeType Complex Type

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

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

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

Child Elements

Element	Type	Description
PullSubscriptionRequest	t:PullSubscriptionRequestType	Specifies a pull subscription type.
PushSubscriptionRequest	t:PushSubscriptionRequestType	Specifies a push subscription type.
StreamingSubscriptionRequest	t:StreamingSubscriptionRequestType	Specifies a subscription to a streaming event notification.<10>

3.1.4.3.2.3   t:BaseSubscriptionRequestType Complex Type

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

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

Child Elements

Element	Type	Description
FolderIds	t:NonEmptyArrayOfBaseFolderIdsType	Specifies an array of folder identifiers that are used to identify folders to

Element	Type	Description
		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.

3.1.4.3.2.4   t:NonEmptyArrayOfNotificationEventTypesType Complex Type

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

```
<xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
  <xs:choice
    maxOccurs="unbounded"
  >
    <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.

3.1.4.3.2.5   t:PullSubscriptionRequestType Complex Type

The PullSubscriptionRequestType complex type specifies a subscription to a pull-based event notification subscription.

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

Child Elements

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

3.1.4.3.2.6 t:PushSubscriptionRequestType Complex Type

The PushSubscriptionRequestType complex type specifies a subscription to a push-based event notification.

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

Child Elements

Element	Type	Description
StatusFrequency	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.

3.1.4.3.3 Elements

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

3.1.4.3.3.1 Subscribe Element

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

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

3.1.4.3.3.2 SubscribeResponse Element

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

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

3.1.4.3.4 Messages

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

3.1.4.3.4.1 tns:SubscribeSoapIn Message

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

Part Name	Element/Type	Description
request	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 3.1.4.3) request.

3.1.4.3.4.2 tns:SubscribeSoapOut Message

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

Part Name	Element/Type	Description
SubscribeResult	tns:SubscribeResponse	Specifies the response message.
ServerVersion	t:ServerVersionInfo	Specifies the schema version for the **Subscribe** operation (section 3.1.4.3) message.

3.1.4.4 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.

3.1.4.4.1 Complex Types

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

3.1.4.4.1.1 m:UnsubscribeResponseType Complex Type

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

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

3.1.4.4.1.2 m:UnsubscribeType Complex Type

The UnsubscribeType complex type specifies a request to end a pull notification subscription.

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

Child Elements

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

3.1.4.4.2 Elements

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

3.1.4.4.2.1   Unsubscribe Element

The Unsubscribe element specifies the properties used to unsubscribe from a subscription.

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

3.1.4.4.2.2   UnsubscribeResponse Element

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

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

3.1.4.4.3   Messages

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

3.1.4.4.3.1   tns:UnsubscribeSoapIn Message

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

Part Name	Element/Type	Description
request	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 3.1.4.4) request.

3.1.4.4.3.2   tns:UnsubscribeSoapOut Message

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

Part Name	Element/Type	Description
UnsubscribeResult	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_SP1"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
         <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"/>
         <xs:include schemaLocation="MS-OXWSNTIF-messages.xsd" />
         <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
         <!-- Add global elements and types from messages.xsd -->
      </xs:schema>
      <xs:schema id="types" elementFormDefault="qualified" version="Exchange2010_SP1"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
         <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
         <!-- Add global elements and types from types.xsd -->
      </xs:schema>
   </wsdl:types>
   <wsdl:portType name="ExchangeServicePortType">
      <wsdl:operation name="Subscribe">
         <wsdl:input message="tns:SubscribeSoapIn" />
         <wsdl:output message="tns:SubscribeSoapOut" />
      </wsdl:operation>
      <wsdl:operation name="Unsubscribe">
```

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

```
                    <soap:header message="tns:GetEventsSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:GetEventsSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:body parts="request" use="literal" />
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="GetEventsResult" use="literal" />
                <soap:header message="tns:GetEventsSoapOut" part="ServerVersion"
use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="GetStreamingEvents">
          <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetEvents" />
          <wsdl:input>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="Impersonation"
use="literal"/>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="MailboxCulture"
use="literal"/>
            <soap:header message="tns:GetStreamingEventsSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
          </wsdl:input>
          <wsdl:output>
            <soap:body parts="GetStreamingEventsResult" use="literal" />
            <soap:header message="tns:GetStreamingEventsSoapOut" part="ServerVersion"
use="literal"/>
          </wsdl:output>
        </wsdl:operation>
      </wsdl:binding>
   <wsdl:message name="SubscribeSoapIn">
      <wsdl:part name="request" element="tns:Subscribe" />
      <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
      <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
      <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
   </wsdl:message>
   <wsdl:message name="SubscribeSoapOut">
      <wsdl:part name="SubscribeResult" element="tns:SubscribeResponse" />
      <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
   </wsdl:message>
   <wsdl:message name="UnsubscribeSoapIn">
      <wsdl:part name="request" element="tns:Unsubscribe" />
      <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
      <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
      <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
   </wsdl:message>
   <wsdl:message name="UnsubscribeSoapOut">
      <wsdl:part name="UnsubscribeResult" element="tns:UnsubscribeResponse" />
      <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
   </wsdl:message>
   <wsdl:message name="GetEventsSoapIn">
      <wsdl:part name="request" element="tns:GetEvents" />
      <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
      <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
      <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
   </wsdl:message>
   <wsdl:message name="GetEventsSoapOut">
      <wsdl:part name="GetEventsResult" element="tns:GetEventsResponse" />
      <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
```

```
    </wsdl:message>
    <wsdl:message name="GetStreamingEventsSoapIn">
      <wsdl:part name="request" element="tns:GetStreamingEvents" />
      <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
      <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
      <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
    </wsdl:message>
    <wsdl:message name="GetStreamingEventsSoapOut">
      <wsdl:part name="GetStreamingEventsResult" element="tns:GetStreamingEventsResponse" />
      <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
    </wsdl:message>
  </wsdl:definitions>
```

6.2 Types Schema

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

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

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

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010_SP1" id="types">
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
   <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
   <xs:complexType name="BaseNotificationEventType">
      <xs:sequence>
         <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="BaseObjectChangedEventType">
      <xs:complexContent>
         <xs:extension base="t:BaseNotificationEventType">
            <xs:sequence>
               <xs:element name="TimeStamp" type="xs:dateTime"/>
               <xs:choice>
                  <xs:element name="FolderId" type="t:FolderIdType"/>
                  <xs:element name="ItemId" type="t:ItemIdType"/>
               </xs:choice>
               <xs:element name="ParentFolderId" type="t:FolderIdType"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="BaseSubscriptionRequestType" abstract="true">
      <xs:sequence>
         <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"
 minOccurs="0"/>
```

```
            <xs:element name="EventTypes" type="t:NonEmptyArrayOfNotificationEventTypesType"/>
            <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
    </xs:complexType>
    <xs:complexType name="ModifiedEventType">
        <xs:complexContent>
            <xs:extension base="t:BaseObjectChangedEventType">
                <xs:sequence>
                    <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="MovedCopiedEventType">
        <xs:complexContent>
            <xs:extension base="t:BaseObjectChangedEventType">
                <xs:sequence>
                    <xs:choice>
                        <xs:element name="OldFolderId" type="t:FolderIdType"/>
                        <xs:element name="OldItemId" type="t:ItemIdType"/>
                    </xs:choice>
                    <xs:element name="OldParentFolderId" type="t:FolderIdType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="NonEmptyArrayOfNotificationEventTypesType">
        <xs:choice maxOccurs="unbounded">
            <xs:element name="EventType" type="t:NotificationEventTypeType"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="NotificationType">
        <xs:sequence>
            <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
            <xs:element name="PreviousWatermark" type="t:WatermarkType" minOccurs="0"/>
            <xs:element name="MoreEvents" type="xs:boolean" minOccurs="0"/>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="CopiedEvent" type="t:MovedCopiedEventType"/>
                <xs:element name="CreatedEvent" type="t:BaseObjectChangedEventType"/>
                <xs:element name="DeletedEvent" type="t:BaseObjectChangedEventType"/>
                <xs:element name="ModifiedEvent" type="t:ModifiedEventType"/>
                <xs:element name="MovedEvent" type="t:MovedCopiedEventType"/>
                <xs:element name="NewMailEvent" type="t:BaseObjectChangedEventType"/>
                <xs:element name="StatusEvent" type="t:BaseNotificationEventType"/>
                <xs:element name="FreeBusyChangedEvent" type="t:BaseObjectChangedEventType"/>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="NotificationEventTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="CopiedEvent"/>
            <xs:enumeration value="CreatedEvent"/>
            <xs:enumeration value="DeletedEvent"/>
            <xs:enumeration value="ModifiedEvent"/>
            <xs:enumeration value="MovedEvent"/>
            <xs:enumeration value="NewMailEvent"/>
            <xs:enumeration value="FreeBusyChangedEvent"/>
        </xs:restriction>
```

```
        </xs:simpleType>
        <xs:complexType name="PushSubscriptionRequestType">
            <xs:complexContent>
                <xs:extension base="t:BaseSubscriptionRequestType">
                    <xs:sequence>
                        <xs:element name="StatusFrequency" type="t:SubscriptionStatusFrequencyType"/>
                        <xs:element name="URL" type="xs:string"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="PullSubscriptionRequestType">
            <xs:complexContent>
                <xs:extension base="t:BaseSubscriptionRequestType">
                    <xs:sequence>
                        <xs:element name="Timeout" type="t:SubscriptionTimeoutType"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:simpleType name="SubscriptionIdType">
            <xs:restriction base="t:NonEmptyStringType"/>
        </xs:simpleType>
        <xs:simpleType name="WatermarkType">
            <xs:restriction base="t:NonEmptyStringType"/>
        </xs:simpleType>
        <xs:simpleType name="SubscriptionTimeoutType">
            <xs:restriction base="xs:int">
                <xs:minInclusive value="1"/>
                <xs:maxInclusive value="1440"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="SubscriptionStatusFrequencyType">
            <xs:restriction base="xs:int">
                <xs:minInclusive value="1"/>
                <xs:maxInclusive value="1440"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="StreamingSubscriptionRequestType">
            <xs:sequence>
                <xs:element name="FolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"
 minOccurs="0"/>
                <xs:element name="EventTypes" type="t:NonEmptyArrayOfNotificationEventTypesType"/>
            </xs:sequence>
            <xs:attribute name="SubscribeToAllFolders" type="xs:boolean" use="optional"/>
        </xs:complexType>
        <xs:complexType name="NonEmptyArrayOfSubscriptionIdsType">
            <xs:sequence>
                <xs:element name="SubscriptionId" type="t:SubscriptionIdType"
 maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="NonEmptyArrayOfNotificationsType">
            <xs:sequence>
                <xs:element name="Notification" type="t:NotificationType" minOccurs="0"
 maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <xs:simpleType name="StreamingSubscriptionConnectionTimeoutType">
```

```
            <xs:restriction base="xs:int">
                <xs:minInclusive value="1"/>
                <xs:maxInclusive value="30"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType name="ConnectionStatusType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="OK"/>
                <xs:enumeration value="Closed"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:schema>
```

6.3 Messages Schema

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

MS-OXWSNTIF-types.xsd includes the 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 6.2
MS-OXWSNTIF-types.xsd	Section 6.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
elementFormDefault="qualified" version="Exchange2010_SP1" id="messages">
    <!--   CHANGE THE SCHEMA LOCATION TO REFLECT THE TYPES XSD ASSOCIATED WITH THIS DOCUMENT
     <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="types.xsd"/>
    <xs:include/>
-->
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSNTIF-types.xsd"/>
    <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
    <xs:complexType name="GetEventsType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:sequence>
                    <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
                    <xs:element name="Watermark" type="t:WatermarkType"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="GetEvents" type="m:GetEventsType"/>
    <xs:complexType name="GetEventsResponseMessageType">
        <xs:complexContent>
            <xs:extension base="m:ResponseMessageType">
```

Release: Thursday, July 29, 2010

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

```
<xs:complexType name="SubscribeResponseMessageType">
    <xs:complexContent>
        <xs:extension base="m:ResponseMessageType">
            <xs:sequence>
                <xs:element name="SubscriptionId" type="t:SubscriptionIdType" minOccurs="0"/>
                <xs:element name="Watermark" type="t:WatermarkType" minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="SubscribeResponseType">
    <xs:complexContent>
        <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
</xs:complexType>
<xs:element name="SubscribeResponse" type="m:SubscribeResponseType"/>
<xs:complexType name="UnsubscribeType">
    <xs:complexContent>
        <xs:extension base="m:BaseRequestType">
            <xs:sequence>
                <xs:element name="SubscriptionId" type="t:SubscriptionIdType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="Unsubscribe" type="m:UnsubscribeType"/>
<xs:complexType name="UnsubscribeResponseType">
    <xs:complexContent>
        <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
</xs:complexType>
<xs:element name="UnsubscribeResponse" type="m:UnsubscribeResponseType"/>
</xs:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Exchange Server 2010

- Microsoft® Exchange Server 2010 Service Pack 1 (SP1)

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. 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 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.

<1> Section 2.2.3.2: GetStreamingEventsResponseMessageType is not part of the schema for Exchange 2010.

<2> Section 2.2.3.4: Exchange 2010 does not include the minOccurs value .

<3> Section 2.2.3.6: In Exchange 2010, the <UnreadCount> element includes a maxOccurs value of 1.

<4> Section 2.2.3.8: Exchange 2010 does not include the minOccurs value for the element..

<5> Section 2.2.3.8: Exchange 2010 does not include the minOccurs value for this element.

<6> Section 2.2.3.8: In Exchange 2010, the <xs:choice> element includes a minOccurs value of 1.

<7> Section 2.2.3.8: Exchange 2010 does not include the <FreeBusyChangedEvent>.

<8> Section 3.1.4.2: Exchange 2010 does not include the **GetStreamingEvents** operation.

<9> Section 3.1.4.3.1.1: Exchange 2010 does not include the **FreeBusyChangedEvent** enumeration value.

<10> Section 3.1.4.3.2.2: Exchange 2010 does not include the <StreamingSubscriptionRequest> element.

<11> Section 3.1.4.3.2.4: In Exchange 2010, the <xs:choice> element has a minOccurs value of 1 .

8   Change Tracking

This section identifies changes that were made to the [MS-OXWSNTIF] protocol document between the May 2010 and August 2010 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.

- An extensive rewrite, addition, or deletion of major portions of content.

- The removal of a document from the documentation set.

- Changes made for template compliance.

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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- 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 protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.2.1 Normative References	55751 Moved [MS-OXGLOS] from the Normative References section to the Informative References section.	N	Content update.
1.2.1 Normative References	Added the reference [MS-OXWSPSNTIF] and removed the references [RFC2396], [RFC2616], and [RFC2818].	N	Content update.
2.2.3.2 m:GetStreamingEventsResponseMessageType Complex Type	56016 Added note about schema version.	N	New product behavior note added.
2.2.3.4 t:BaseNotificationEventType Complex Type	56016 Added a product behavior note that states that Exchange 2010 does not include the minOccurs value.	N	New product behavior note added.
2.2.3.6 t:ModifiedEventType Complex Type	Removed the maxOccurs value from the <UnreadCount> element.	N	Content update.
2.2.3.8 t:NotificationType Complex Type	56016 Added product behavior notes	N	New product

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
	that state that Exchange 2010 does not include the minOccurs value for the PreviousWatermark and MoreEvents elements, and the choice element.		behavior note added.
2.2.3.8 t:NotificationType Complex Type	56016 Added a product behavior note that states that Exchange 2010 does not include the FreeBusyChangedEvent element.	N	New product behavior note added.
3.1.4.2 GetStreamingEvents	56018 Added a product behavior note that states that the GetStreamingEvents operation is not included in Exchange 2010.	N	New product behavior note added.
3.1.4.3.1.1 t:NotificationEventTypeType Simple Type	56018 Added a product behavior note that states that the FreeBusyChangedEvent enumeration value is not included in Exchange 2010.	N	New product behavior note added.
3.1.4.3.2.2 m:SubscribeType Complex Type	56016 Added a product behavior note that states that the <StreamingSubscriptionRequest> element is not implemented in Exchange 2010.	N	New product behavior note added.
3.1.4.3.2.4 t:NonEmptyArrayOfNotificationEventTypesType Complex Type	Removed the minOccurs value from the element syntax.	N	Content update.
3.1.4.3.2.4 t:NonEmptyArrayOfNotificationEventTypesType Complex Type	56018 Added a product behavior note that states that the xs:choice element in Exchange 2010 includes the minOccurs value.	N	New product behavior note added.
6.3 Messages Schema	56427 Added another file to the list of files in the table.	N	Content update.
7 Appendix B: Product Behavior	56663 Removed "Beta" from Exchange Server 2010 SP1 product name.	N	Content update.

9 Index

A

Applicability 8

C

Capability negotiation 8
Change tracking 47
Client
   abstract data model 34
   initialization 34
   local events 34
   message processing 34
   sequencing rules 34
   timer events 34
   timers 34

F

Full WSDL 37

G

Glossary 6

I

Introduction 6

M

Messages
   syntax 9
   transport 9

O

Overview (synopsis) 7

P

Preconditions 8
Prerequisites 8
Product behavior 46

R

References
   informative 7
   normative 6
Relationship to other protocols 7

S

Security
   implementer considerations 36
   parameter index 36
Server

   abstract data model 17
   initialization 17
   local events 34
   message processing 17
   sequencing rules 17
   timer events 34
   timers 17
Standards assignments 8

T

Tracking changes 47

V

Vendor-extensible fields 8
Versioning 8