

[MS-OXWSPSNTIF]: Push 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 iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

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	1.2.0	Minor	Updated the technical content.
05/05/2010	1.2.1	Editorial	Revised and edited the technical content.
08/04/2010	2.0	Major	Significantly changed the technical content.
11/03/2010	3.0	Major	Significantly changed the technical content.

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 Overview	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	6
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
2.3 Namespaces	8
2.4 Simple Types	8
2.5 Complex Types	8
2.6 Elements	8
2.7 Attributes	8
2.8 Groups	8
2.9 Attribute Groups	9
2.10 Messages	9
3 Protocol Details	10
3.1 ExchangeServicePortType Server Details	10
3.1.1 Abstract Data Model	10
3.1.2 Timers	10
3.1.3 Initialization	10
3.1.4 Message Processing Events and Sequencing	10
3.1.4.1 SendNotification	10
3.1.4.1.1 Simple Types	11
3.1.4.1.1.1 t:SubscriptionStatusType Simple Type	11
3.1.4.1.2 Complex Types	11
3.1.4.1.2.1 m:SendNotificationResponseMessageType Complex Type	11
3.1.4.1.2.2 m:SendNotificationResponseType Complex Type	12
3.1.4.1.2.3 m:SendNotificationResultType Complex Type	12
3.1.4.1.3 Elements	12
3.1.4.1.3.1 SendNotification Element	13
3.1.4.1.3.2 SendNotificationResult Element	13
3.1.4.1.4 Messages	13
3.1.4.1.4.1 tns:SendNotificationSoapIn Message	13
3.1.4.1.4.2 tns:SendNotificationSoapOut Message	13
3.1.5 Timer Events	13
3.1.6 Other Local Events	13
3.2 Client Details	14
3.2.1 Abstract Data Model	14
3.2.2 Timers	14
3.2.3 Initialization	14

3.2.4	Message Processing Events and Sequencing	14
3.2.5	Timer Events	14
3.2.6	Other Local Events	14
4	Protocol Examples	15
5	Security	16
5.1	Security Considerations for Implementers	16
5.2	Index of Security Parameters	16
6	Appendix A: Full WSDL	17
6.1	WSDL	17
6.2	Types Schema	18
6.3	Messages Schema	19
7	Appendix B: Product Behavior	21
8	Change Tracking	22
9	Index	24

1 Introduction

This document specifies the Push Notifications Web Service protocol, which is responsible for receiving subscribed events that are sent by the server.

1.1 Glossary

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

Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
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-OXWSNTIF] Microsoft Corporation, "[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>

[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 Push Notifications Web Service protocol provides clients with subscribed event updates that are sent by the server. Clients subscribe to notifications, as specified in [\[MS-OXWSNTIF\]](#), creating a push subscription that tells the server where to send notifications. The client then creates a Web service that enables it to receive the notifications that are sent by the server.

1.4 Relationship to Other Protocols

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

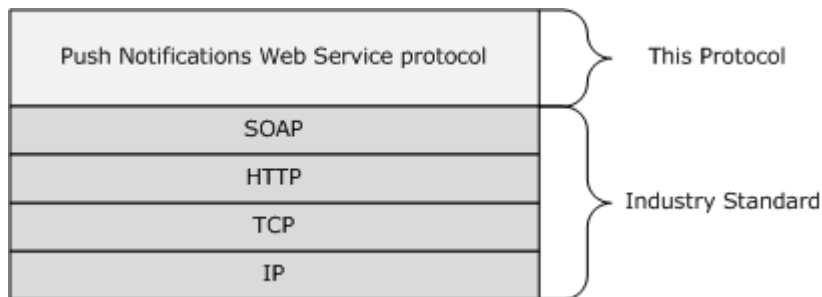


Figure 1: Push Notification Web Service protocol HTTP stack

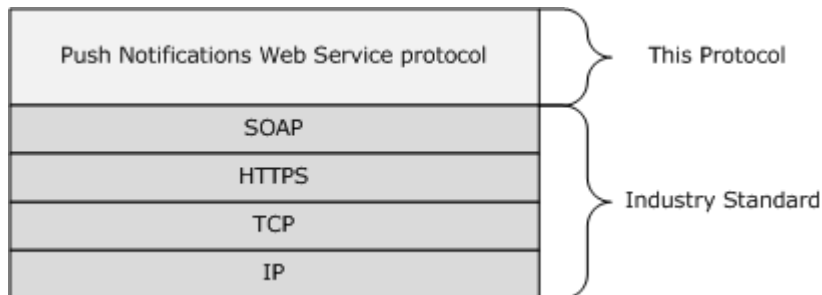


Figure 2: Push Notification Web Service protocol HTTPS stack

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The protocol specified in this document is applicable to tightly coupled client-server environments in which the client and the server are always connected. This protocol is not applicable to environments in which the client periodically connects to the server. If clients are loosely coupled with the server, the Notifications Web Service protocol as specified in [\[MS-OXWSNTIF\]](#), is applicable.

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 [\[WSDL\]](#).

2.3 Namespaces

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

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

2.4 Simple Types

This specification does not define any common **XML schema** simple type definitions.

2.5 Complex Types

This specification does not define any common **XML schema** complex type definitions.

2.6 Elements

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

2.7 Attributes

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

2.8 Groups

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

2.9 Attribute Groups

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

2.10 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 that are returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 ExchangeServicePortType Server Details

The Push Notifications Web Service protocol defines a single port type.

Operation	Description
SendNotification	Sends subscription information from the server to the client application.

3.1.1 Abstract Data Model

The Push 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 operation listed in the following table.

Operation	Description
SendNotification (section 3.1.4.1)	Sends subscription information from the server to the client application.

3.1.4.1 SendNotification

The [SendNotification](#) operation sends subscription information from the server to the client application.

Request

Message Format	Description
tns:SendNotificationSoapIn	Specifies the SOAP message that requests the notification.

Response

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

3.1.4.1.1 Simple Types

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

3.1.4.1.1.1 t:SubscriptionStatusType Simple Type

The [SubscriptionStatusType](#) simple type specifies the status of a push subscription.

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

Enumeration

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

Value	Description
OK	Specifies that the server will continue to send notifications.
Unsubscribe	Specifies that the server will stop sending notifications and end the subscription.

3.1.4.1.2 Complex Types

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

3.1.4.1.2.1 m:SendNotificationResponseMessageType Complex Type

The **SendNotificationResponseMessageType** complex type specifies the status and result of a single [SendNotification](#) operation (section [3.1.4.1](#)) request. The **SendNotificationResponseMessageType** complex type extends the [ResponseMessageType](#) complex type ([\[MS-OXWSCDATA\]](#) section 2.2.3.52).

```
<xs:complexType name="SendNotificationResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Notification"
          type="t:NotificationType"
          minOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

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

```

Child Elements

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

3.1.4.1.2.2 m:SendNotificationResponseType Complex Type

The **SendNotificationResponseType** complex type specifies the push notifications that are sent by the server to the client application. The **SendNotificationResponseType** complex type extends the [BaseResponseMessageType](#) complex type ([MS-OXWSCDATA] section 2.2.3.15).

```

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

```

3.1.4.1.2.3 m:SendNotificationResultType Complex Type

The [SendNotificationResultType](#) complex type specifies the response of a client application to a push notification.

```

<xs:complexType name="SendNotificationResultType">
  <xs:sequence>
    <xs:element name="SubscriptionStatus"
      type="t:SubscriptionStatusType"
    />
  </xs:sequence>
</xs:complexType>

```

Child Elements

Element	Type	Description
SubscriptionStatus	t:SubscriptionStatusType	Specifies the status of a push subscription.

3.1.4.1.3 Elements

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

3.1.4.1.3.1 SendNotification Element

The [SendNotification](#) element specifies the push notifications that are sent by the server to the client application.

```
<xs:element name="SendNotification"
  type="m:SendNotificationResponseType"
/>
```

3.1.4.1.3.2 SendNotificationResult Element

The [tns:SendNotificationResult](#) element specifies the response of a client application to a push notification.

```
<xs:element name="SendNotificationResult"
  type="m:SendNotificationResultType"
/>
```

3.1.4.1.4 Messages

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

3.1.4.1.4.1 tns:SendNotificationSoapIn Message

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

Part Name	Element/Type	Description
request	tns:SendNotification	Specifies the request.
RequestVersion	t:RequestServerVersion	Specifies the schema version for the SendNotification operation (section 3.1.4.1.3.1) request.

3.1.4.1.4.2 tns:SendNotificationSoapOut Message

The SendNotificationSoapOut message contains one part, as described in the following table.

Part Name	Element/Type	Description
SendNotificationResult	tns:SendNotificationResult	Specifies the response message.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

3.2 Client 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 that are returned by the transport are passed directly back to the higher-layer protocol or application.

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 Push 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-OXWSPSNTIF.wsdl	Contains the WSDL for the implementation of this protocol.	6.1
MS-OXWSPSNTIF-types.xsd	Contains the XML schema type definitions that are used in this protocol.	6.2
MS-OXWSPSNTIF-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-OXWSPSNTIF-types.xsd or MS-OXWSPSNTIF-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

6.1 WSDL

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

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:s="http://www.w3.org/2001/XMLSchema"

  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import
        namespace="http://schemas.microsoft.com/exchange/services/2006/messages" schemaLocation="MS-
        OXWSPSNTIF-messages.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="SendNotificationSoapIn">
    <wsdl:part name="request" element="tns:SendNotification" />
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
  </wsdl:message>
  <wsdl:message name="SendNotificationSoapOut">
    <wsdl:part name="SendNotificationResult" element="tns:SendNotificationResult" />
  </wsdl:message>
  <wsdl:portType name="NotificationServicePortType">
    <wsdl:operation name="SendNotification">
      <wsdl:input message="tns:SendNotificationSoapIn" />
      <wsdl:output message="tns:SendNotificationSoapOut" />
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="NotificationServiceBinding" type="tns:NotificationServicePortType">
    <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 transport="http://schemas.xmlsoap.org/soap/http" style="document" />

    <wsdl:operation name="SendNotification">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/SendNotification" />
        <wsdl:input>
            <soap:header message="tns:SendNotificationSoapIn" part="RequestVersion"
use="literal"/>
            <soap:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>

</wsdl:binding>

<wsdl:binding name="NotificationServiceBinding12" type="tns:NotificationServicePortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />

    <wsdl:operation name="SendNotification">
        <soap12:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/SendNotification" />
        <wsdl:input>
            <soap12:header message="tns:SendNotificationSoapIn" part="RequestVersion"
use="literal"/>
            <soap12:body parts="request" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal" />
        </wsdl:output>
    </wsdl:operation>

</wsdl:binding>

</wsdl:definitions>

```

6.2 Types Schema

This section contains the contents of the MS-OXWSPSNTIF-types.xsd file.

```

<?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:simpleType name="SubscriptionStatusType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="Unsubscribe"/>
        </xs:restriction>

```

```

    </xs:simpleType>
</xs:schema>

```

6.3 Messages Schema

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

MS-OXWSPSNTIF-types.xsd includes the files listed in the following table. To operate correctly, these files have to be 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
MS-OXWSNTIF-types.xsd	[MS-OXWSNTIF] section 6

```

<?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:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
  schemaLocation="MS-OXWSPSNTIF-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:complexType name="SendNotificationResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="Notification" type="t:NotificationType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="SendNotificationResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="SendNotification" type="m:SendNotificationResponseType"/>
  <xs:complexType name="SendNotificationResultType">
    <xs:sequence>
      <xs:element name="SubscriptionStatus" type="t:SubscriptionStatusType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SendNotificationResult" type="m:SendNotificationResultType"/>

```

</xs:schema>

7 Appendix B: Product Behavior

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

- Microsoft® Exchange Server 2007
- 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. 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.

8 Change Tracking

This section identifies changes that were made to the [MS-OXWSPSNTIF] protocol document between the August 2010 and November 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.
- Changes made for template compliance.
- Removal of a document from the documentation set.

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

The revision class **Editorial** means that the 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 revision type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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
Z Appendix B: Product Behavior	57843 Added Exchange 2007 to the list of applicable products.	Y	Content updated.

9 Index

A

[Applicability](#) 7

C

[Capability negotiation](#) 7

[Change tracking](#) 22

Client

[abstract data model](#) 14

[initialization](#) 14

[local events](#) 14

[message processing](#) 14

[overview](#) 14

[sequencing rules](#) 14

[timer events](#) 14

[timers](#) 14

F

[Full WSDL](#) 17

G

[Glossary](#) 5

I

[Introduction](#) 5

M

Messages

[overview](#) 8

[syntax](#) 8

[transport](#) 8

O

[Overview \(synopsis\)](#) 6

P

[Preconditions](#) 6

[Prerequisites](#) 6

[Product behavior](#) 21

R

[Relationship to other protocols](#) 6

S

Security

[implementer considerations](#) 16

[overview](#) 16

[parameter index](#) 16

Server

[abstract data model](#) 10

[initialization](#) 10

[local events](#) 13

[message processing](#) 10

[overview](#) 10

[sequencing rules](#) 10

[timer events](#) 13

[timers](#) 10

[Standards assignments](#) 7

T

[Tracking changes](#) 22

V

[Vendor-extensible fields](#) 7

[Versioning](#) 7