[MS-OXWSPSNTIF]:

Push Notifications Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	1.2.0	Minor	Updated the technical content.
5/5/2010	1.2.1	Editorial	Revised and edited the technical content.
8/4/2010	1.3	Minor	Clarified the meaning of the technical content.
11/3/2010	2.0	Major	Significantly changed the technical content.
3/18/2011	2.1	Minor	Clarified the meaning of the technical content.
8/5/2011	3.0	Major	Significantly changed the technical content.
10/7/2011	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	4.0	Major	Significantly changed the technical content.
4/27/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	4.1	Minor	Clarified the meaning of the technical content.
2/11/2013	4.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	4.1	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	4.2	Minor	Clarified the meaning of the technical content.
2/10/2014	4.2	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	4.3	Minor	Clarified the meaning of the technical content.
7/31/2014	4.4	Minor	Clarified the meaning of the technical content.
10/30/2014	4.4	No Change	No changes to the meaning, language, or formatting of the technical content.
5/26/2015	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction 6
1.1 Glossary 6
1.2 References 7
1.2.1 Normative References 7
1.2.2 Informative References 8
1.3 Overview 8
1.4 Relationship to Other Protocols 8
1.5 Prerequisites/Preconditions 9
1.6 Applicability Statement 9
1.7 Versioning and Capability Negotiation 9
1.8 Vendor-Extensible Fields 9
1.9 Standards Assignments 9

2 Messages 10
2.1 Transport 10
2.2 Common Message Syntax 10
2.2.1 Namespaces 10
2.2.2 Messages 10
2.2.3 Elements 11
2.2.4 Complex Types 11
2.2.5 Simple Types 11
2.2.6 Attributes 11
2.2.7 Groups 11
2.2.8 Attribute Groups 11

3 Protocol Details 12
3.1 ExchangeServicePortType Server Details 12
3.1.1 Abstract Data Model 12
3.1.2 Timers 12
3.1.3 Initialization 12
3.1.4 Message Processing Events and Sequencing Rules 12
3.1.4.1 SendNotification Operation 12
3.1.4.1.1 Messages 13
3.1.4.1.1.1 tns:SendNotificationSoapIn Message 13
3.1.4.1.1.2 tns:SendNotificationSoapOut Message 13
3.1.4.1.2 Elements 14
3.1.4.1.2.1 SendNotification Element 14
3.1.4.1.2.2 SendNotificationResult Element 14
3.1.4.1.3 Complex Types 14
3.1.4.1.3.1 m:SendNotificationResponseMessageType Complex Type 15
3.1.4.1.3.2 m:SendNotificationResponseType Complex Type 15
3.1.4.1.3.3 m:SendNotificationResultType Complex Type 16
3.1.4.1.4 Simple Types 16
3.1.4.1.4.1 t:SubscriptionStatusType Simple Type 16
3.1.4.1.5 Attributes 17
3.1.4.1.6 Groups 17
3.1.4.1.7 Attribute Groups 17
3.1.5 Timer Events 17
3.1.6 Other Local Events 17

4 Protocol Examples 18

5 Security 19
5.1 Security Considerations for Implementers 19
5.2 Index of Security Parameters 19

6 Appendix A: Full WSDL 20
7 Appendix B: Full XML Schema 22
7.1 Messages Schema 22
7.2 Types Schema 23
8 Appendix C: Product Behavior 24
9 Change Tracking 25
10 Index 27

1 Introduction

The Push Notifications Web Service Protocol enables clients to receive subscribed event updates sent by the server.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS): An extension of **HTTP** that securely encrypts and decrypts webpage requests.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [SOAP1.1] section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a SOAP message to its recipient. See [SOAP1.2-1/2007] section 5.3 for more information.

SOAP header: A mechanism for implementing extensions to a SOAP message in a decentralized manner without prior agreement between the communicating parties. See [SOAP1.2-1/2007] section 5.2 for more information.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

web server: A server computer that hosts websites and responds to requests from applications.

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a WSDL operation [WSDL]. Also, an element that describes the data being exchanged between web service providers and clients.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [RFC3986]. A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [XMLNS-2ED].

XML namespace prefix: An abbreviated form of an **XML namespace**, as described in [XML].

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by XML itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

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

[MS-OXWSNTIF] Microsoft Corporation, "Notifications Web Service Protocol".

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

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

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

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

[WSDLSOAP] Angelov, D., Ballinger, K., Butek, R., et al., "WSDL 1.1 Binding Extension for SOAP 1.2", W3C Member Submission, April 2006, http://www.w3.org/Submission/2006/SUBM-wsdl11soap12-20060405/

[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

[WSIBASIC] Ballinger, K., Ehnebuske, D., Gudgin, M., et al., Eds., "Basic Profile Version 1.0", Final Material, April 2004, http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

[XMLSCHEMA1] Thompson, H., 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.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

[MS-OXWSADISC] Microsoft Corporation, "Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol".

1.3 Overview

This protocol provides clients with subscribed event updates that are sent by the server. Clients subscribe to these notifications, as described in [MS-OXWSNTIF], by creating a push subscription that specifies where the server is to send notifications. The clients then create a web service that enables them to receive the notifications sent to them by the server.

1.4 Relationship to Other Protocols

A client that implements this protocol can use the Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], to identify the target **endpoint** to use for each operation.

This protocol uses the Simple Object Access Protocol (SOAP) Protocol, as described in [SOAP1.1], to specify the structure information exchanged between the client and server. This protocol uses the XML Protocol, as described in [XMLSCHEMA1] and [XMLSCHEMA2], to describe the message content sent to and from the server.

This protocol uses **SOAP** over HTTP, as described in [RFC2616], and SOAP over **HTTPS**, as described in [RFC2818], as shown in the following layering diagram.

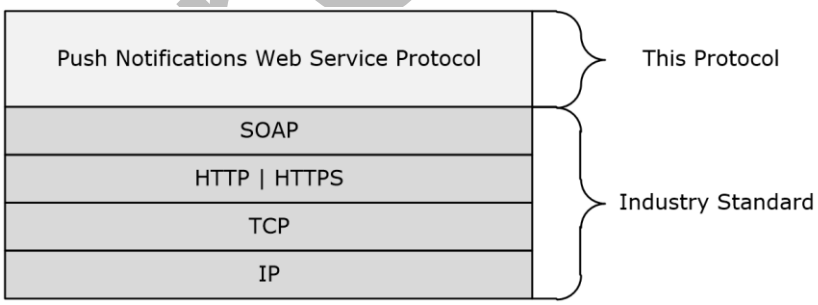

Figure 1: This protocol in relation to other protocols

The notification information that is sent to this protocol is used when requests are made by using the Notifications Web Service Protocol [MS-OXWSNTIF].

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

The endpoint **URL** that is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as described in [MS-OXWSADISC], or the Autodiscover Publishing and Lookup Protocol, as described in [MS-OXDSCLI], is required to form the **HTTP** request to the **web server** that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP web requests that target this protocol.

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 connects to the server periodically. If clients are loosely coupled with the server, the Notifications Web Service Protocol, as described 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 multiple transports with SOAP 1.1, as specified in section 2.1 and in [SOAP1.1].
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [MS-OXWSCDATA] section 2.2.3.11.
- **Security and Authentication Methods:** This protocol relies on the web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section 3.1.4.
- **Capability Negotiation**: This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, or **present**.

2.1 Transport

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

This protocol relies on the web server that hosts the application to perform authentication. This protocol MUST support SOAP over HTTP, as specified in [RFC2616]. The protocol SHOULD use secure communications by means of HTTPS, as defined in [RFC2818].

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML 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	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	
m	http://schemas.microsoft.com/exchange/services/2006/messages	
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[WSDLSOAP]
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
wsi	http://ws-i.org/schemas/conformanceClaim/	[WSIBASIC]

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

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

2.2.4 Complex Types

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

This protocol defines a single port type with one operation.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the WSDL operations as defined by this specification.

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

3.1.4.1 SendNotification Operation

The **SendNotification** operation sends subscription information from the server to the client application.

The following is the WSDL port type specification for the **SendNotification** operation.

```
<wsdl:operation name="SendNotification">
   <wsdl:input message="tns:SendNotificationSoapIn" />
   <wsdl:output message="tns:SendNotificationSoapOut" />
</wsdl:operation>
```

The following is the WSDL binding specification for the **SendNotification** operation.

```
<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>
```

3.1.4.1.1 Messages

The following table lists and describes the WSDL message definitions that are specific to the **SendNotification** operation.

Message name	Description
SendNotificationSoapIn	Specifies a request to the **SendNotification** operation.
SendNotificationSoapOut	Specifies a response from the **SendNotification** operation.

3.1.4.1.1.1 tns:SendNotificationSoapIn Message

The **SendNotificationSoapIn** WSDL message specifies the **SendNotification** operation request to send a notification to the client application.

```
<wsdl:message name="SendNotificationSoapIn">
        <wsdl:part name="request" element="tns:SendNotification" />
        <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

The **SendNotificationSoapIn** WSDL message is the input message for the **SOAP action** http://schemas.microsoft.com/exchange/services/2006/messages/SendNotification.

The parts of the **SendNotificationSoapIn** WSDL message are listed and described in the following table.

Part name	Element/type	Description
request	**tns:SendNotification** (section 3.1.4.1.2.1)	Specifies the **SOAP body** of the request.
RequestVersion	**t:RequestServerVersion** ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a **SOAP header** that identifies the schema version for the **SendNotification** operation request.

3.1.4.1.1.2 tns:SendNotificationSoapOut Message

The **SendNotificationSoapOut** WSDL message specifies the response to the **SendNotification** operation request.

```
<wsdl:message name="SendNotificationSoapOut">
        <wsdl:part name="SendNotificationResult" element="tns:SendNotificationResult" />
</wsdl:message>
```

The **SendNotificationSoapOut** WSDL message is the output message for the SOAP action http://schemas.microsoft.com/exchange/services/2006/messages/SendNotification.

The parts of the **SendNotificationSoapOut** WSDL message are listed and described in the following table.

Part name	Element/type	Description
SendNotificationResult	**tns:SendNotificationResult** (section 3.1.4.1.2.2)	Specifies SOAP body of the response message.

3.1.4.1.2 Elements

The following table lists and describes the XML schema element definitions that are specific to the **SendNotification** operation.

Element name	Description
SendNotification	Specifies the push notifications that are sent by the server to the client application.
SendNotificationResult	Specifies the response of a client application to a push notification.

3.1.4.1.2.1 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"
 />
```

3.1.4.1.2.2 SendNotificationResult Element

The **SendNotificationResul**t element specifies the response of a client application to a push notification.

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

3.1.4.1.3 Complex Types

The following table lists and describes the XML schema complex type definitions that are specific to the **SendNotification** operation.

Complex type name	Description
SendNotificationResponseMessageType	Specifies the status and result of a single **SendNotification** operation request.

Complex type name	Description
SendNotificationResponseType	Specifies the push notifications that are sent by the server to the client application.
SendNotificationResultType	Specifies the response of a client application to a push notification.

3.1.4.1.3.1 m:SendNotificationResponseMessageType Complex Type

The **SendNotificationResponseMessageType** complex type specifies the status and result of a single **SendNotification** operation request. The **SendNotificationResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.65.

```
<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>
```

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

Element name	Type	Description
Notification	**t:NotificationType** ([MS-OXWSNTIF] section 2.2.4.8)	Specifies the subscription and the events that have occurred since the last notification.

3.1.4.1.3.2 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, as specified in [MS-OXWSCDATA] section 2.2.4.18.

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

3.1.4.1.3.3 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>
```

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

Element name	Type	Description
SubscriptionStatus	**t:SubscriptionStatusType** (section 3.1.4.1.4.1)	Specifies the status of a push subscription.

3.1.4.1.4 Simple Types

The following table lists and describes the XML schema simple type definitions that are specific to the **SendNotification** operation.

Simple type name	Description
SubscriptionStatusType	Specifies the status type of a push subscription.

3.1.4.1.4.1 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>
```

The following table lists and describes the values that are defined by the **SubscriptionStatusType** simple type.

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

3.1.4.1.5 Attributes

None.

3.1.4.1.6 Groups

None.

3.1.4.1.7 Attribute Groups

None.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality described in this document.

File name	Description	Section
MS-OXWSPSNTIF.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSPSNTIF-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	7.1
MS-OXWSPSNTIF-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

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

This section provides 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>
```

7 Appendix B: Full XML Schema

For ease of implementation, the following sections provide the full XML schema for this protocol.

Schema name	Prefix	Section
Messages schema	m:	7.1
Types schema	t:	7.2

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

7.1 Messages Schema

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

MS-OXWSPSNTIF-messages.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 7.1
MS-OXWSNTIF-types.xsd	[MS-OXWSNTIF] section 7.2

```
<?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="Exchange2016" id="messages">
     <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>
```

7.2 Types Schema

This section provides 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/messages"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2016" 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>
```

8 Appendix C: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016 Preview

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

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

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

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

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
8 Appendix C: Product Behavior	Added Exchange 2016 to list of supported products.	Y	Content update.

10 Index

A

Abstract data model
  server 12
Applicability 9
Attribute groups 11
Attributes 11

C

Capability negotiation 9
Change tracking 25
Complex types 11

D

Data model - abstract
  server 12

E

Events
  local - server 17
  timer - server 17
Examples 18

F

Fields - vendor-extensible 9
Full WSDL 20
Full XML schema 22
  Messages Schema 22
  Types Schema 23

G

Glossary 6
Groups 11

I

Implementer - security considerations 19
Index of security parameters 19
Informative references 8
Initialization
  server 12
Introduction 6

L

Local events
  server 17

M

Message processing
  server 12
Messages
  attribute groups 11
  attributes 11
  complex types 11
  elements 11
  enumerated 10
  groups 11
  namespaces 10
  simple types 11
  syntax 10
  transport 10

N

Namespaces 10
Normative references 7

O

Operations
  SendNotification Operation 12
Overview (synopsis) 8

P

Parameters - security index 19
Preconditions 9
Prerequisites 9
Product behavior 24
Protocol Details
  overview 12

R

References 7
  informative 8
  normative 7
Relationship to other protocols 8

S

Security
  implementer considerations 19
  parameter index 19
Sequencing rules
  server 12
Server
  abstract data model 12
  initialization 12
  local events 17
  message processing 12
  SendNotification Operation operation 12
  sequencing rules 12
  timer events 17
  timers 12
Simple types 11
Standards assignments 9
Syntax
  messages - overview 10

T

Timer events
server 17
Timers
server 12
Tracking changes 25
Transport 10
Types
complex 11
simple 11

V

Vendor-extensible fields 9
Versioning 9

W

WSDL 20

X

XML schema 22
Messages Schema 22
Types Schema 23