[MS-ALERTSS]:
Alerts Service Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	3.0	Major	Significantly changed the technical content.
04/11/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
09/12/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	3.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

This document specifies the Alerts Service Protocol. The Alerts Service Protocol allows a protocol client to list and delete alert subscriptions. Alert subscriptions specify when and how notifications are sent to users when changes are made to content stored on the server. The protocol does not specify the creation or editing of alert subscriptions.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

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

GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
XML

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

alert
alert subscription
authenticated user
current user
document
item
list
list item
Simple Object Access Protocol (SOAP)
site
SOAP action
SOAP body
SOAP fault
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
web discussion comment
Web Services Description Language (WSDL)
WSDL operation
WSDL port type
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

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

1.2.1 Normative References

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

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, November 1987, http://www.ietf.org/rfc/rfc1035.txt

[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

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

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

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

[RFC2822] Resnick, P., Ed., "Internet Message Format", STD 11, RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.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/

[SOAP1.2/1] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624

[SOAP1.2/2] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., and Nielsen, H.F., "SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624

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

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

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

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

1.2.2 Informative References

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

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[MS-OSALER] Microsoft Corporation, "Alerts Interoperability Protocol Specification".

1.3 Protocol Overview (Synopsis)

The protocol allows a protocol client to list and delete existing **alert subscriptions**. It consists of a single **WSDL port type** with two **WSDL operations** and their replies. It describes a series of communications between the protocol client and protocol server roles.

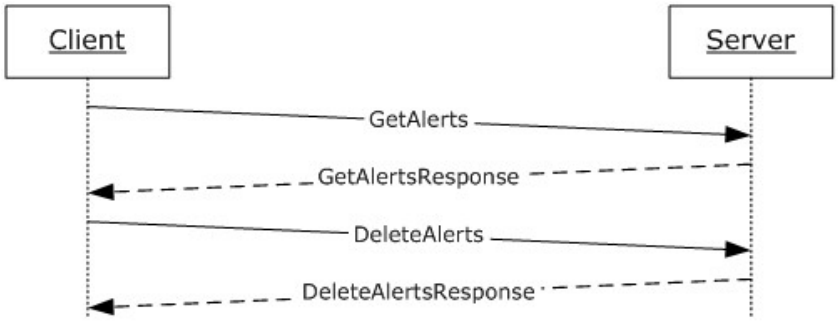

Figure 1: Protocol client/server message exchange

Prior to the initiation of the protocol, the protocol client is configured with user information. After this is done, the client connects by using the user information. For more information, see section 1.5.

1.3.1 Alert Enumeration

The protocol client can now initiate an exchange by requesting a list of alert subscriptions for the specified user. The server responds with a list of metadata about the alert subscriptions to which the **authenticated user** has subscribed.

1.3.2 Alert Deletion

The protocol client can delete an alert subscription by sending a request to the protocol server that contains a list of **GUIDs** for the alert subscriptions to delete. The protocol server responds with information about deletion(s) that failed. The protocol server will no longer send the alert subscriptions for the GUIDs that did not fail.

1.4 Relationship to Other Protocols

This protocol uses the **SOAP** message protocol for formatting request and response messages, as described in [SOAP1.1], [SOAP1.2/1] and [SOAP1.2/2]. It transmits those messages by using **HTTP**, as described in [RFC2616], or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, as described in [RFC2818].

The following diagram shows the underlying messaging and transport stack used by the protocol.

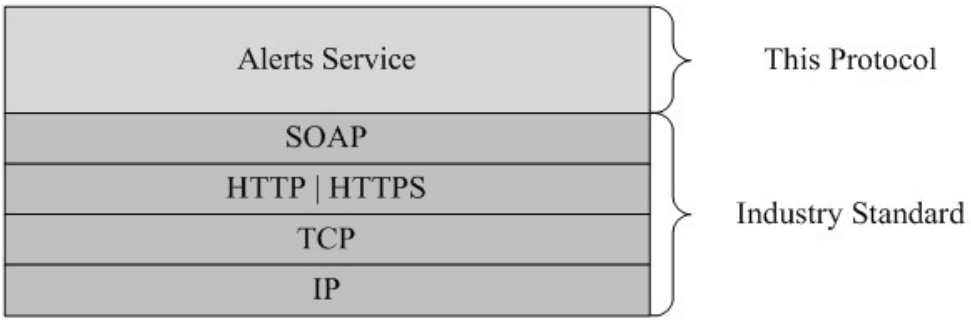

Figure 2: This protocol in relation to other protocols

The protocol has no interactions with parallel protocols, nor are there other protocols that substitute for it.

1.5 Prerequisites/Preconditions

This protocol operates against a **site (2)** that is identified by a **URL** that is known by protocol clients. The protocol server endpoint is formed by appending "/_vti_bin/Alerts.asmx" to the URL of the site (2), for example, http://www.contoso.com/Repository/_vti_bin/Alerts.asmx.

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

This protocol is used to assist in the management of alert subscriptions. The service allows a protocol client to get the list of alert subscriptions for a particular authenticated user. It also allows a protocol client to delete alert subscriptions.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP as specified in section 2.1.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol servers MUST support SOAP over HTTP. Protocol servers SHOULD additionally support SOAP over HTTPS, as specified in [RFC2818], for securing communication with protocol clients.

Protocol messages MUST be formatted as specified in [SOAP1.1], section 4, "SOAP Envelope", or in [SOAP1.2/1], section 5, "SOAP Message Construct". Protocol server faults MUST be returned either by using HTTP Status Codes as specified in [RFC2616], section 10, "Status Code Definitions", or by using **SOAP faults** as specified in [SOAP1.1], section 4.4, "SOAP Fault" or in [SOAP1.2/1], section 5.4, "SOAP Fault".

2.2 Common Message Syntax

This section contains common definitions used by this protocol. The syntax of the definitions uses **XML schema** as specified in [XMLSCHEMA1] and [XMLSCHEMA2], and **Web Services Description Language (WSDL)** as specified 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/sharepoint/soap/2002/1/alerts/	
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]
soap12	http://schemas.xmlsoap.org/wsdl/soap12/	[SOAP1.2/1] [SOAP1.2/2]
(none)	http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]

2.2.2 Messages

None.

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 a pass-through. No additional timers or other states are 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.

This protocol allows protocol servers to notify protocol clients of application-level faults by using SOAP faults. Except where specified, these SOAP faults are not significant for interoperability, and protocol clients can interpret them in an implementation-specific manner.

This protocol allows protocol servers to perform implementation-specific authorization checks and notify protocol clients of authorization faults either by using HTTP Status Codes or by using SOAP faults as specified previously in this section.

3.1 Alerts Service Protocol Server Details

3.1.1 Abstract Data Model

This section describes a conceptual model of data organization that an implementation maintains to participate in this protocol. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with the behavior described in this document.

Information about notifications and alert subscriptions is specified in [MS-OSALER].

This protocol retrieves and deletes alert subscriptions. Each authenticated user will have an individual list of alert subscriptions.

An alert subscription is a persistent request on the server for a notification (**alert (1)**) that a particular **document** or **list (1)** has been modified in a specific way. The alert subscription contains the information that determines when to generate a notification, and how to deliver the notification.

3.1.2 Timers

None.

3.1.3 Initialization

The initial protocol state is provided to the protocol by user authentication and connection to its server.

3.1.4 Message Processing Events and Sequencing Rules

The following table specifies the WSDL operations provided by the protocol.

Operation	Description
DeleteAlerts	Deletes alert subscriptions from a user's current list of alert subscriptions. Returns a **DeleteAlertsResponse**. If the server cannot locate an alert subscription using an identifier from the List, or cannot properly parse the identifier, it skips that identifier without notifying the client.
GetAlerts	Enumerates the current alert subscriptions for an authenticated user. It returns a **GetAlertsResponse**.

The protocol client is not required to perform a **GetAlerts** request prior to submitting a **DeleteAlerts** request. However, using the **GetAlerts** operation is the typical method for acquiring valid **AlertID**s to provide to the **DeleteAlerts** operation.

3.1.4.1 DeleteAlerts

This operation deletes alert subscriptions from a user's current list of alert subscriptions.

```
<wsdl:operation name="DeleteAlerts">
    <wsdl:input message="tns:DeleteAlertsSoapIn" />
    <wsdl:output message="tns:DeleteAlertsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **DeleteAlertsSoapIn** request message and the protocol server responds with a **DeleteAlertsSoapOut** response message.

The protocol client provides a list of alert identifiers in the **DeleteAlertsSoapIn** message. The protocol server deletes each alert subscription on the protocol server that matches one of the provided identifiers. The protocol server then responds with information about any deletions that failed. The response does not include information about successful deletions.

Deletion errors are counted during this operation. The maximum number of errors is implementation-dependent<1>. If the maximum number of errors is reached, processing is terminated and a **DeleteFailure** error message that contains the **TooManyErrors** error type is appended to the response message.

If the string representing an identifier is not in the proper format, that identifier is ignored in the **DeleteAlertsSoapOut** response message, but the error is still counted toward the total number of deletion errors.

3.1.4.1.1 Messages

3.1.4.1.1.1 DeleteAlertsSoapIn

This message contains the request to begin a **DeleteAlerts** WSDL operation.

The **SOAP action** value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/DeleteAlerts
```

The **SOAP body** contains a **DeleteAlerts** element.

3.1.4.1.1.2 DeleteAlertsSoapOut

This message represents the response associated with the **DeleteAlerts** WSDL operation. It contains the status for any part of the **DeleteAlerts** WSDL operation that failed. If the reply is empty, there were no failures of the error type specified in section 3.1.4.1.4.1.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/DeleteAlerts
```

The SOAP body contains a **DeleteAlertsResponse** element.

3.1.4.1.2 Elements

3.1.4.1.2.1 DeleteAlerts

This structure is contained in a **DeleteAlertsSoapIn** message and contains the list of GUIDs that the client is requesting to be deleted.

```
<s:element name="DeleteAlerts">
  <s:complexType>
    <s:sequence>
      <s:element name="IDs" type="tns:ArrayOfString" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

IDs: This is an **ArrayOfString** structure with one string for each alert subscription to be deleted. The string MUST be formatted as a GUID.

3.1.4.1.2.2 DeleteAlertsResponse

This structure is the content of a **DeleteAlertsSoapOut** message. It contains a list of **DeleteAlertsResult** structures, with at least one for each GUID in the **DeleteAlertsRequest** that failed during processing.

```
<s:element name="DeleteAlertsResponse">
  <s:complexType>
    <s:sequence>
      <s:element name="DeleteAlertsResult"
         type="tns:ArrayOfDeleteFailureDefinition" minOccurs="0" maxOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

DeleteAlertsResult: A structure that contains the information about a single failure that occurred during processing of a **DeleteAlerts** operation.

3.1.4.1.3 Complex Types

3.1.4.1.3.1 ArrayOfString

In this protocol, an **ArrayOfString** structure is used to pass multiple strings, each of which represents a GUID associated with a specific alert subscription. Each string in the sequence MUST be formatted as a GUID.

```
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element name="string" type="s:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

string: Each string in the array represents a single GUID which references a particular alert subscription.

3.1.4.1.3.2 ArrayOfDeleteFailureDefinition

This structure is a list of **DeleteFailure** structures, each of which represents a single failure while processing a **DeleteAlerts** operation.

```
<s:complexType name="ArrayOfDeleteFailureDefinition">
  <s:sequence>
    <s:element name="DeleteFailure"
       type="tns:DeleteFailureDefinition" minOccurs="0"
       maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

DeleteFailure: A single **DeleteFailureDefinition** structure, as specified in section 3.1.4.1.3.3.

3.1.4.1.3.3 DeleteFailureDefinition

This structure represents the error message associated with a failure that occurred while processing one **GUID** from a **DeleteAlerts** operation. The structure contains the GUID that failed as a string, and an **ErrorType** indicating why the failure occurred.

```
<s:complexType name="DeleteFailureDefinition">
  <s:sequence>
    <s:element name="ID" type="s:string" minOccurs="0"/>
    <s:element name="Error" type="tns:ErrorType"/>
  </s:sequence>
</s:complexType>
```

ID: A string representing the GUID that failed. The string MUST be formatted as a GUID. This element is not returned when **Error** is **TooManyErrors**.

Error: The **ErrorType** indicating why the deletion failed.

3.1.4.1.4 Simple Types

3.1.4.1.4.1 ErrorType

ErrorType describes a particular failure that occurred during **DeleteAlerts** request processing.

```
<s:simpleType name="ErrorType">
  <s:restriction base="s:string">
    <s:enumeration value="None"/>
    <s:enumeration value="AccessDenied"/>
    <s:enumeration value="ServerError"/>
    <s:enumeration value="TooManyErrors"/>
  </s:restriction>
</s:simpleType>
```

The following table defines the allowable values. The protocol server MUST NOT return a value of "None".

Value	Meaning
None	The alert subscription was successfully deleted.
AccessDenied	An authenticated user is not authorized to delete the specified alert subscription.<2>
ServerError	An unknown or implementation-specific protocol server error occurred during a **DeleteAlerts** operation.
TooManyErrors	The maximum number of failures occurred while the protocol server was processing a **DeleteAlerts** request message.

3.1.4.2 GetAlerts

This operation lists the current alert subscriptions for an authenticated user.

```
<wsdl:operation name="GetAlerts">
    <wsdl:input message="tns:GetAlertsSoapIn" />
    <wsdl:output message="tns:GetAlertsSoapOut" />
</wsdl:operation>
```

The protocol client sends a **GetAlertsSoapIn** request message and the protocol server responds with a **GetAlertsSoapOut** response message.

The **GetAlertsSoapOut** response message contains common information and a list of information about each alert subscription.

3.1.4.2.1 Messages

3.1.4.2.1.1 GetAlertsSoapIn

The message represents the protocol client request for the current alert subscriptions from the protocol server.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/GetAlerts
```

The SOAP body contains a **GetAlerts** element.

3.1.4.2.1.2 GetAlertsSoapOut

The message represents the protocol server response for a protocol client request for the list of current alert subscriptions.

The SOAP action value of the message is defined as:

```
http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/GetAlerts
```

The SOAP body contains a **GetAlertsResponse** element.

3.1.4.2.2 Elements

3.1.4.2.2.1 GetAlerts

GetAlerts is an empty structure indicating that a **GetAlerts** WSDL operation is being requested.

```
<s:element name="GetAlerts">
  <s:complexType/>
</s:element>
```

3.1.4.2.2.2 GetAlertsResponse

This structure contains the response to a **GetAlerts** WSDL operation.

```
<s:element name="GetAlertsResponse">
  <s:complexType>
    <s:sequence>
        <s:element name="GetAlertsResult" type="tns:AlertInfoDefinition"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

GetAlertsResult: A structure of type **AlertInfoDefinition**, as specified in **AlertInfoDefinition** (section 3.1.4.2.3.1).

3.1.4.2.3 Complex Types

3.1.4.2.3.1 AlertInfoDefinition

The **AlertInfoDefinition** structure contains common information for the authenticated user listed as **current user**, in addition to a list of alert subscriptions.

```
<s:complexType name="AlertInfoDefinition">
  <s:sequence>
    <s:element name="CurrentUser" type="s:string" minOccurs="0"/>
    <s:element name="AlertServerName" type="s:string" minOccurs="0"/>
    <s:element name="AlertServerUrl" type="s:string" minOccurs="0"/>
    <s:element name="AlertServerType" type="s:string" minOccurs="0"/>
    <s:element name="AlertsManagementUrl" type="s:string" minOccurs="0"/>
    <s:element name="AlertWebTitle" type="s:string" minOccurs="0"/>
    <s:element name="NewAlertUrl" type="s:string" minOccurs="0"/>
    <s:element name="AlertWebId" type="s:string" minOccurs="0"/>
    <s:element name="Alerts" type="tns:ArrayOfAlertDefinition"
       minOccurs="0"/>
  </s:sequence>
</s:complexType>
```

CurrentUser: The display name of the authenticated user for the current connection.

AlertServerName: The name of the server storing the current list of alert subscriptions. This name MUST conform to "Domain Names- Implementation and Specification", [RFC1035] section 2.3.1, and "Uniform Resource Identifiers (URI): Generic Syntax", [RFC2396]. This is the host component of the request URL.

AlertServerUrl: A **URI** for the site (2) storing the alert subscriptions. This MUST be the site (2) URL.

AlertServerType: A string identifying the server software storing the alert subscriptions.<3>

AlertsManagementUrl: A URL to a page for the current list of alert subscriptions. This MUST reference a Web page that an authenticated user can request that summarizes the user's current alert subscriptions and allows for their management.

AlertWebTitle: The title of the site (2) on which the alert subscriptions are stored.

NewAlertUrl: A URL that represents a site (2) page that allows an authenticated user to create new alert subscriptions.

AlertWebId: The identifier of the site (2) on which the alert subscriptions are stored. This MUST contain a string representing a GUID that uniquely identifies the site containing the returned alert subscriptions.

Alerts: The alert subscriptions. This represents a list of alert subscription structures that contain information about each current alert subscription.

3.1.4.2.3.2 ArrayOfAlertDefinition

This structure contains a list of zero or more alert subscription structures.

```
<s:complexType name="ArrayOfAlertDefinition">
  <s:sequence>
    <s:element name="Alert" type="tns:Alert"
       minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

Alert: A single **Alert** subscription structure, as specified in section 3.1.4.2.3.3.

3.1.4.2.3.3 Alert

This structure represents a single alert subscription.

```
<s:complexType name="Alert">
  <s:sequence>
    <s:element name="Id" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="Title" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="Active" type="s:boolean" minOccurs="1" maxOccurs="1"/>
    <s:element name="EventType" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="AlertForTitle" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="AlertForUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="EditAlertUrl" type="s:string" minOccurs="0" maxOccurs="1"/>
    <s:element name="DeliveryChannels"
       type="tns:ArrayOfDeliveryChannelDefinition" minOccurs="0" maxOccurs="1"/>
  </s:sequence>
</s:complexType>
```

Id: The identifier of an alert subscription. This MUST contain a string representing a GUID which uniquely identifies the alert subscription.

Title: A string specifying the friendly name of the alert subscription.

Active: A BOOLEAN value indicating whether this alert subscription will generate alerts. This MUST contain the value "true".

EventType: A string indicating the type of event that caused the alert subscription to generate a notification. This MUST be a value from the following table.

Value	Description
Add	Triggered when a new **item** is added.
Modify	Triggered when an item is modified.
Delete	Triggered when an item is deleted.
Discussion	Triggered when a **Web discussion comment** associated with an item is added, modified, or deleted.
All	Triggered when any of the previous values is triggered.

AlertForTitle: A string specifying the title of the resource, such as a list (1) or a **list item** being monitored by the alert subscription.

AlertForUrl: The URL of the resource, such as a list (1) or a list item being monitored. This MUST be an absolute URL.

EditAlertUrl: The URL referencing a Web page to be used to edit the alert subscription. This MUST be an absolute URL.

DeliveryChannels: A list of the methods used to provide notifications when an alert subscription is triggered.

3.1.4.2.3.4 ArrayOfDeliveryChannelDefinition

This structure contains a list of zero or more **DeliveryChannel** structures.

```
<s:complexType name="ArrayOfDeliveryChannelDefinition">
  <s:sequence>
    <s:element name="DeliveryChannel" type="tns:DeliveryChannel"
       nillable="true" minOccurs="0" maxOccurs="unbounded"/>
  </s:sequence>
</s:complexType>
```

DeliveryChannel: See **DeliveryChannel** (section 3.1.4.2.3.5).

3.1.4.2.3.5 DeliveryChannel

A structure describing how a notification is to be delivered.

```
<s:complexType name="DeliveryChannel" abstract="true"/>
```

3.1.4.2.3.6 EmailChannel

This structure is used to provide the information required to deliver a notification through an electronic mail service.

```
<s:complexType name="EmailChannel">
  <s:complexContent mixed="false">
    <s:extension base="tns:DeliveryChannel">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Frequency" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Address" type="s:string" />
      </s:sequence>
    </s:extension>
  </s:complexContent>
</s:complexType>
```

Frequency: A string indicating how often the notification is sent out. The value MUST be listed in the following table.

Value	Description
Immediate	A notification is sent immediately.
Daily	A notification is sent during daily processing.
Weekly	A notification is sent during weekly processing.

Address: This is the destination for notifications. It is an Internet Message Format-compliant [RFC2822] string to be used in the recipient field of a message header, as specified in [RFC2822].

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The overall scenario is that a protocol client requires the list of alert subscriptions for the current authenticated user. The client submits a **GetAlerts** request similar to the following code example.

```
<?xml version="1.0"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope">
 <Body>
  <GetAlerts
    xmlns="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts">
  </GetAlerts>
 </Body>
</Envelope>
```

The protocol server receives this request and creates an appropriate response, similar to the following code example.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <GetAlertsResponse
    xmlns="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts">
      <GetAlertsResult>
        <CurrentUser>Auricchio, Jose Luis</CurrentUser>
        <AlertServerName>widgets</AlertServerName>
        <AlertServerUrl>http://widgets</AlertServerUrl>
        <AlertServerType>STS</AlertServerType>
        <AlertsManagementUrl>
          http://widgets/_layouts/MySubs.aspx
        </AlertsManagementUrl>
        <AlertWebTitle>widgets</AlertWebTitle>
        <NewAlertUrl>http://widgets/_layouts/SubChoos.aspx</NewAlertUrl>
        <AlertWebId>4a3d9478-00a0-4ea8-bdb6-36034a189433</AlertWebId>
        <Alerts>
          <Alert>
            <Id>{76061063-9C09-4C4D-B1A1-16D3F0CDF1F8}</Id>
            <Title>Shared Documents</Title>
            <Active>true</Active>
            <EventType>All</EventType>
            <AlertForUrl>http://widgets/Shared Documents</AlertForUrl>
            <EditAlertUrl>http://widgets/_layouts/SubEdit.aspx?Alert={76061063-9C09-4C4D-
B1A1-16D3F0CDF1F8}&amp;amp;List={071C725D-EF5E-4779-B95C-0F8173C260E1}</EditAlertUrl>
            <DeliveryChannels>
              <DeliveryChannel xsi:type="EmailChannel">
                <Frequency>Immediate</Frequency>
                <Address>jose@cohowinery.com</Address>
              </DeliveryChannel>
            </DeliveryChannels>
          </Alert>
        </Alerts>
      </GetAlertsResult>
    </GetAlertsResponse>
  </soap:Body>
</soap:Envelope>
```

If the user directs the protocol client to delete an existing alert subscription, the request would resemble the following code example.

```
<?xml version="1.0"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope">
 <Body>
  <DeleteAlerts
  xmlns="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts">
   <IDs>
    <string>{76061063-9C09-4C4D-B1A1-16D3F0CDF1F8}</string>
   </IDs>
  </DeleteAlerts>
 </Body>
</Envelope>
```

The protocol server response would typically resemble the following code example.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteAlertsResponse
    xmlns="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts">
      <DeleteAlertsResult />
    </DeleteAlertsResponse>
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

For ease of implementation, the full WSDL is provided here.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/">
      <s:element name="GetAlerts">
        <s:complexType />
      </s:element>
      <s:element name="GetAlertsResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="GetAlertsResult" type="tns:AlertInfoDefinition"/>
          </s:sequence>
        </s:complexType>
      </s:element>
       <s:complexType name="AlertInfoDefinition">
         <s:sequence>
            <s:element name="CurrentUser" type="s:string" minOccurs="0"/>
            <s:element name="AlertServerName" type="s:string" minOccurs="0"/>
            <s:element name="AlertServerUrl" type="s:string" minOccurs="0"/>
            <s:element name="AlertServerType" type="s:string" minOccurs="0"/>
            <s:element name="AlertsManagementUrl" type="s:string" minOccurs="0"/>
            <s:element name="AlertWebTitle" type="s:string" minOccurs="0"/>
            <s:element name="NewAlertUrl" type="s:string" minOccurs="0"/>
            <s:element name="AlertWebId" type="s:string" minOccurs="0"/>
            <s:element name="Alerts" type="tns:ArrayOfAlertDefinition" minOccurs="0"/>
         </s:sequence></s:complexType>
       <s:complexType name="ArrayOfAlertDefinition">
          <s:sequence>
            <s:element name="Alert" type="tns:Alert" minOccurs="0" maxOccurs="unbounded"/>
          </s:sequence></s:complexType>
      <s:complexType name="Alert">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
               name="Id" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
               name="Title" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1"
               name="Active" type="s:boolean" />
          <s:element minOccurs="0" maxOccurs="1"
               name="EventType" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
               name="AlertForTitle" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
               name="AlertForUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
               name="EditAlertUrl" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1"
               name="DeliveryChannels"
               type="tns:ArrayOfDeliveryChannelDefinition" />
```

```
      </s:sequence>
    </s:complexType>
    <s:complexType name="ArrayOfDeliveryChannelDefinition">
      <s:sequence>
        <s:element name="DeliveryChannel" type="tns:DeliveryChannel"
          nillable="true" minOccurs="0" maxOccurs="unbounded"/>
      </s:sequence>
    </s:complexType>
    <s:complexType name="DeliveryChannel" abstract="true" />
    <s:complexType name="EmailChannel">
      <s:complexContent mixed="false">
        <s:extension base="tns:DeliveryChannel">
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1"
                name="Frequency" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1"
                name="Address" type="s:string" />
          </s:sequence>
        </s:extension>
      </s:complexContent>
    </s:complexType>
    <s:element name="DeleteAlerts">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
              name="IDs" type="tns:ArrayOfString" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfString">
      <s:sequence>
        <s:element name="string" type="s:string" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
      </s:sequence>
    </s:complexType>
    <s:element name="DeleteAlertsResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1"
              name="DeleteAlertsResult"
             type="tns:ArrayOfDeleteFailureDefinition" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="ArrayOfDeleteFailureDefinition">
      <s:sequence>
        <s:element name="DeleteFailure"
           type="tns:DeleteFailureDefinition" minOccurs="0"
           maxOccurs="unbounded"/>
      </s:sequence>
    </s:complexType>
    <s:complexType name="DeleteFailureDefinition">
      <s:sequence>
        <s:element name="ID" type="s:string" minOccurs="0"/>
        <s:element name="Error" type="tns:ErrorType"/>
      </s:sequence>
    </s:complexType>
    <s:simpleType name="ErrorType">
      <s:restriction base="s:string">
```

```
          <s:enumeration value="None" />
          <s:enumeration value="AccessDenied" />
          <s:enumeration value="ServerError" />
          <s:enumeration value="TooManyErrors" />
        </s:restriction>
      </s:simpleType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="GetAlertsSoapIn">
    <wsdl:part name="parameters" element="tns:GetAlerts" />
  </wsdl:message>
  <wsdl:message name="GetAlertsSoapOut">
    <wsdl:part name="parameters" element="tns:GetAlertsResponse" />
  </wsdl:message>
  <wsdl:message name="DeleteAlertsSoapIn">
    <wsdl:part name="parameters" element="tns:DeleteAlerts" />
  </wsdl:message>
  <wsdl:message name="DeleteAlertsSoapOut">
    <wsdl:part name="parameters" element="tns:DeleteAlertsResponse" />
  </wsdl:message>
  <wsdl:portType name="AlertsSoap">
    <wsdl:operation name="GetAlerts">
      <wsdl:input message="tns:GetAlertsSoapIn" />
      <wsdl:output message="tns:GetAlertsSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="DeleteAlerts">
      <wsdl:input message="tns:DeleteAlertsSoapIn" />
      <wsdl:output message="tns:DeleteAlertsSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="AlertsSoap" type="tns:AlertsSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetAlerts">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/GetAlerts"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteAlerts">
      <soap:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/DeleteAlerts"
style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="AlertsSoap12" type="tns:AlertsSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="GetAlerts">
```

```
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/GetAlerts"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteAlerts">
      <soap12:operation
soapAction="http://schemas.microsoft.com/sharepoint/soap/2002/1/alerts/DeleteAlerts"
style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

7 Appendix B: 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® Office Outlook® 2007
- Microsoft® SharePoint® Foundation 2010
- Windows® SharePoint® Services 2.0
- Windows® SharePoint® Services 3.0
- Microsoft® SharePoint® Foundation 2013

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

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

<1> Section 3.1.4.1: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and SharePoint Foundation 2010 limit errors in the processing of **DeleteAlerts** to 20 errors in all released versions. The protocol client and protocol server are not dependent on this particular behavior.

<2> Section 3.1.4.1.4.1: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, SharePoint Foundation 2010, and SharePoint Foundation 2013 never return this value.

<3> Section 3.1.4.2.3.1: Windows SharePoint Services 2.0, Windows SharePoint Services 3.0, and SharePoint Foundation 2010 set this value to "STS".

8 Change Tracking

This section identifies changes that were made to the [MS-ALERTSS] protocol document between the September 2012 and October 2012 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
1.3.1 Alert Enumeration	Renamed the authenticated users as authenticated user.	N	Content updated.
1.5 Prerequisites/Preconditions	Added definition number for term "site".	N	Content updated.
1.6 Applicability Statement	Renamed the authenticated users as authenticated user.	N	Content updated.
3.1.1 Abstract Data Model	Renamed the authenticated users as authenticated user.	N	Content updated.
3.1.4 Message Processing Events and Sequencing Rules	Renamed the authenticated users as authenticated user.	N	Content updated.
3.1.4.1 DeleteAlerts	Added the namespace prefix to schema fragment.	N	New content added.
3.1.4.1.4.1 ErrorType	Added a product behavior note to specify behavior for the AccessDenied error.	N	New product behavior note added.
3.1.4.1.4.1 ErrorType	Renamed the authenticated users as authenticated user.	N	Content updated.
3.1.4.2 GetAlerts	Added the namespace prefix to schema fragment.	N	New content added.
3.1.4.2	Renamed the authenticated users	N	Content updated.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
GetAlerts	as authenticated user.		
3.1.4.2.3.1 AlertInfoDefinition	Added the appropriate number for site.	N	Content updated.
3.1.4.2.3.1 AlertInfoDefinition	Renamed the authenticated users as authenticated user.	N	Content updated.
4 Protocol Examples	Renamed the authenticated users as authenticated user.	N	Content updated.

9 Index

T

V

W