

[MS-OXWSMTRK]: Message Tracking Web Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
07/15/2009	1.0	Major	Initial Availability.
11/04/2009	1.1.0	Minor	Updated the technical content.
02/10/2010	1.2.0	Minor	Updated the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary.....	5
1.2 References.....	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Protocol Overview	6
1.4 Relationship to Other Protocols.....	6
1.5 Prerequisites/Preconditions.....	7
1.6 Applicability Statement.....	7
1.7 Versioning and Capability Negotiation.....	7
1.8 Vendor Extensible Fields.....	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport.....	8
2.2 Common Message Syntax.....	8
2.2.1 Namespaces.....	8
2.2.2 Simple Types	8
2.2.2.1 t:MessageTrackingScopeType Simple Type	8
2.2.3 Complex Types.....	9
2.2.4 Elements.....	9
2.2.5 Attributes.....	9
2.2.6 Groups.....	9
2.2.7 Attribute Groups	9
2.2.8 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 FindMessageTrackingReport	10
3.1.4.1.1 Complex Types.....	11
3.1.4.1.1.1 m:FindMessageTrackingReportRequestType Complex Type	11
3.1.4.1.1.2 m:FindMessageTrackingReportResponseMessageType Complex Type	12
3.1.4.1.1.3 t:ArrayOfFindMessageTrackingSearchResultType Complex Type	13
3.1.4.1.1.4 t:FindMessageTrackingSearchResultType Complex Type	14
3.1.4.1.2 Elements.....	15
3.1.4.1.2.1 FindMessageTrackingReport Element.....	15
3.1.4.1.2.2 FindMessageTrackingReportResponse Element.....	15
3.1.4.1.3 Messages	15
3.1.4.1.3.1 tns:FindMessageTrackingReportSoapIn Message	15
3.1.4.1.3.2 tns:FindMessageTrackingReportSoapOut Message	16
3.1.4.2 GetMessageTrackingReport	16
3.1.4.2.1 Simple Types	16
3.1.4.2.1.1 t:MessageTrackingDeliveryStatusType Simple Type	16
3.1.4.2.1.2 t:MessageTrackingEventDescriptionType Simple Type.....	17
3.1.4.2.1.3 t:MessageTrackingReportTemplateType Simple Type	20
3.1.4.2.2 Complex Types.....	21

3.1.4.2.2.1	m:GetMessageTrackingReportRequestType Complex Type	21
3.1.4.2.2.2	m:GetMessageTrackingReportResponseMessageType Complex Type	22
3.1.4.2.2.3	t:ArrayOfRecipientTrackingEventType Complex Type	23
3.1.4.2.2.4	t:MessageTrackingReportType Complex Type	23
3.1.4.2.2.5	t:RecipientTrackingEventType Complex Type	24
3.1.4.2.3	Elements.....	25
3.1.4.2.3.1	GetMessageTrackingReport Element	26
3.1.4.2.3.2	GetMessageTrackingReportResponse Element.....	26
3.1.4.2.4	Messages	26
3.1.4.2.4.1	tns:GetMessageTrackingReportSoapIn Message	26
3.1.4.2.4.2	tns:GetMessageTrackingReportSoapOut Message	26
3.1.5	Timer Events.....	26
3.1.6	Other Local Events	27
3.2	Client Details.....	27
3.2.1	Abstract Data Model.....	27
3.2.2	Timers	27
3.2.3	Initialization	27
3.2.4	Message Processing Events and Sequencing	27
3.2.5	Timer Events.....	27
3.2.6	Other Local Events	27
4	Protocol Examples	28
5	Security.....	29
5.1	Security Considerations for Implementers.....	29
5.2	Index of Security Parameters	29
6	Appendix A: Full WSDL	30
6.1	WSDL.....	30
6.2	Types Schema.....	32
6.3	Messages Schema	34
7	Appendix B: Product Behavior	36
8	Change Tracking	37
9	Index.....	39

1 Introduction

This document specifies the Message Tracking Web Service protocol, which is responsible for finding and returning information about message delivery by the server. Clients use the SOAP protocol [[SOAP1.1](#)] to contact the message tracking service.

1.1 Glossary

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

blind carbon copy (BCC) recipient
distribution list
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Socket Layers (HTTPS)
mailbox
Simple Mail Transfer Protocol (SMTP)
SOAP body
SOAP fault
SOAP header
SOAP message
Web Services Description Language (WSDL)
WSDL message
WSDL port type
XML
XML namespace
XML schema

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

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

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

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

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

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

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

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

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

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

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

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

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

1.2.2 Informative References

None.

1.3 Protocol Overview

The Message Tracking Web Service protocol provides clients with message delivery information on the server. Clients search for a message on the server by using the [FindMessageTrackingReport](#) operation (section [3.1.4.1](#)), and can then retrieve information from that report by using the [GetMessageTrackingReport](#) operation (section [3.1.4.2](#)).

1.4 Relationship to Other Protocols

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

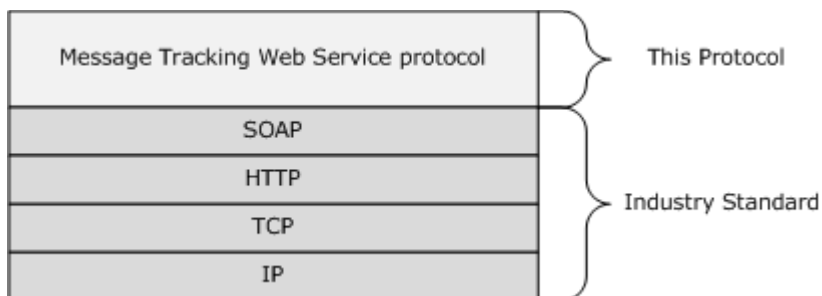


Figure 1: Message Tracking Web Service protocol HTTP stack

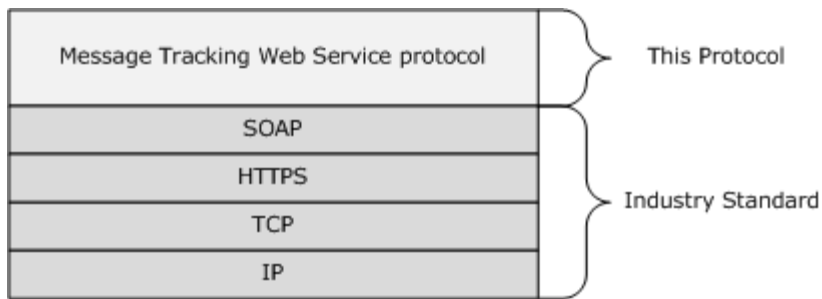


Figure 2: Message Tracking Web Service protocol HTTPS stack

The information that is returned by the Message Tracking Web Service protocol is not used by any additional protocols.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The protocol that is specified in this document is applicable to environments that track message delivery.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).

Capability Negotiation: None.

1.8 Vendor Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific XML 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-OXWSMTRK]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSMTRK]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	[MS-OXWSMTRK]

2.2.2 Simple Types

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

Simple type	Description
t:MessageTrackingScopeType	Specifies where to search for tracking reports.

2.2.2.1 t:MessageTrackingScopeType Simple Type

The [MessageTrackingScopeType](#) simple type specifies where to search for tracking reports.

```
<xs:simpleType name="MessageTrackingScopeType">
  <xs:restriction
    base="xs:string"
  >
```



```

    <xs:enumeration
      value="Organization"
    />
    <xs:enumeration
      value="Forest"
    />
    <xs:enumeration
      value="Site"
    />
  </xs:restriction>
</xs:simpleType>

```

Enumeration

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

Value	Description
Organization	Specifies that the search will cover all forests in the organization.
Forest	Specifies that the search will cover all servers in the forest.
Site	Specifies that the search will cover the local site of the server that executes it.

2.2.3 Complex Types

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

2.2.4 Elements

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

2.2.5 Attributes

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

2.2.6 Groups

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

2.2.7 Attribute Groups

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

2.2.8 Messages

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

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

The Notifications Web Service protocol defines a single port type.

Operation	Description
FindMessageTrackingReport	Finds messages that meet the specified criteria.
GetMessageTrackingReport	Gets tracking information about the specified messages.

3.1.1 Abstract Data Model

The Message Tracking Web Service protocol is a stateless protocol.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing

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

Operation	Description
FindMessageTrackingReport (section 3.1.4.1)	Finds messages that meet the specified criteria.
GetMessageTrackingReport (section 3.1.4.2)	Gets tracking information about the specified messages.

3.1.4.1 FindMessageTrackingReport

The [FindMessageTrackingReport](#) operation finds messages that meet the specified criteria.

```
<wsdl:operation name="FindMessageTrackingReport"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:FindMessageTrackingReportSoapIn" />
  <wsdl:output message="tns:FindMessageTrackingReportSoapOut" />
</wsdl:operation>
```

Request

Message Format	Description
tns:FindMessageTrackingReportSoapIn	Specifies the SOAP message that requests the report.

Response

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

3.1.4.1.1 Complex Types

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

3.1.4.1.1.1 m:FindMessageTrackingReportRequestType Complex Type

The **FindMessageTrackingReportRequestType** complex type specifies the criteria for finding a tracking report. The **FindMessageTrackingReportRequestType** complex type extends the **BaseRequestType** complex type ([\[MS-OXWSCDATA\]](#) section 2.2.3.14).

```
<xs:complexType name="FindMessageTrackingReportRequestType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:all>
        <xs:element name="Scope"
          type="t:MessageTrackingScopeType"
        />
        <xs:element name="Domain"
          type="t:NonEmptyStringType"
        />
        <xs:element name="Sender"
          type="t:EmailAddressType"
          minOccurs="0"
        />
        <xs:element name="PurportedSender"
          type="t:EmailAddressType"
          maxOccurs="0"
        />
        <xs:element name="Recipient"
          type="t:EmailAddressType"
          minOccurs="0"
        />
        <xs:element name="Subject"
          type="xs:string"
          minOccurs="0"
        />
        <xs:element name="StartDateTime"
          type="xs:dateTime"
          minOccurs="0"
        />
        <xs:element name="EndDateTime"
          type="xs:dateTime"
          minOccurs="0"
        />
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

    />
    <xs:element name="MessageId"
      type="t:NonEmptyStringType"
      minOccurs="0"
    />
    <xs:element name="FederatedDeliveryMailbox"
      type="t:EmailAddressType"
      minOccurs="0"
    />
    <xs:element name="DiagnosticsLevel"
      type="xs:string"
      minOccurs="0"
    />
  </xs:all>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Child Elements

Element	Type	Description
Scope	t:MessageTrackingScopeType	Specifies where to perform the search.
Domain	t:NonEmptyStringType	Specifies the domain to search for.
Sender	t:EmailAddressType	Specifies the e-mail address of the person who is sending the message.
PurportedSender	t:EmailAddressType	Specifies the e-mail address of the intended person sending the message.
Recipient	t:EmailAddressType	Specifies the e-mail addresses of the people who are receiving the message.
Subject	xs:string	Specifies the subject filter to search for.
StartDateTime	xs:dateTime	Specifies the search time period, indicating that the message was sent after this value.
EndDateTime	xs:dateTime	Specifies the search time period, indicating that the message was sent before this value.
MessageId	t:NonEmptyStringType	Specifies message identification to search for.
FederatedDeliveryMailbox	t:EmailAddressType	Specifies the mailbox to which a cross-premise message was sent.
DiagnosticsLevel	xs:string	Specifies how detailed the tracing report should be.

3.1.4.1.1.2 m:FindMessageTrackingReportResponseMessageType Complex Type

The **FindMessageTrackingReportResponseMessageType** complex type specifies the response for the **FindMessageTrackingReport** operation (section [3.1.4.1](#)). The

FindMessageTrackingReportResponseMessageType complex type extends the **ResponseMessageType** complex type ([\[MS-OXWSCDATA\]](#) section 2.2.3.52).

```
<xs:complexType name="FindMessageTrackingReportResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Diagnostics"
          type="t:ArrayOfStringsType"
          minOccurs="0"
        />
        <xs:element name="MessageTrackingSearchResults"
          type="t:ArrayOfFindMessageTrackingSearchResultType"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element	Type	Description
Diagnostics	t:ArrayOfStringsType	Specifies timing and performance information that will be used to produce various statistical reports.
MessageTrackingSearchResults	t:ArrayOfFindMessageTrackingSearchResultType	Specifies an array of matching records.

3.1.4.1.1.3 t:ArrayOfFindMessageTrackingSearchResultType Complex Type

The [ArrayOfFindMessageTrackingSearchResultType](#) complex type specifies an array of search results.

```
<xs:complexType name="ArrayOfFindMessageTrackingSearchResultType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="MessageTrackingSearchResult"
      type="t:FindMessageTrackingSearchResultType"
    />
  </xs:choice>
</xs:complexType>
```

Child Elements

Element	Type	Description
MessageTrackingSearchResult	t:FindMessageTrackingSearchResultType	Specifies a message that was found by using the FindMessageTrackingReport operation (section 3.1.4.1).

3.1.4.1.1.4 t:FindMessageTrackingSearchResultType Complex Type

The [FindMessageTrackingSearchResultType](#) complex type specifies information about the message that was found by using the [FindMessageTrackingReport](#) operation (section [3.1.4.1](#)).

```
<xs:complexType name="FindMessageTrackingSearchResultType">
  <xs:all>
    <xs:element name="Subject"
      type="xs:string"
    />
    <xs:element name="Sender"
      type="t:EmailAddressType"
    />
    <xs:element name="PurportedSender"
      type="t:EmailAddressType"
      minOccurs="0"
    />
    <xs:element name="Recipients"
      type="t:ArrayOfRecipientsType"
    />
    <xs:element name="SubmittedTime"
      type="xs:dateTime"
    />
    <xs:element name="MessageTrackingReportId"
      type="t:NonEmptyStringType"
    />
    <xs:element name="PreviousHopServer"
      type="t:NonEmptyStringType"
      minOccurs="0"
    />
  </xs:all>
</xs:complexType>
```

Child Elements

Element	Type	Description
Subject	xs:string	Specifies the subject of the message that was found.
Sender	t:EmailAddressType	Specifies the e-mail address of the sender for the message that was found.
PurportedSender	t:EmailAddressType	Specifies the e-mail address of the intended person sending the message.
Recipients	t:ArrayOfRecipientsType	Specifies the e-mail addresses of the recipients for

Element	Type	Description
		the message that was found.
SubmittedTime	xs:dateTime	Specifies the time that the message entered the server.
MessageTrackingReportId	t:NonEmptyStringType	Specifies the message by its message-id, the organization where the message was found, the server on which the message was submitted, and an internal ID that uniquely identifies the message.
PreviousHopServer	t:NonEmptyStringType	Specifies the previous server name, when available, that submitted the message.

3.1.4.1.2 Elements

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

3.1.4.1.2.1 FindMessageTrackingReport Element

The [FindMessageTrackingReport](#) element specifies a request that includes the search criteria to identify a message to track.

```
<xs:element name="FindMessageTrackingReport"
  type="m:FindMessageTrackingReportRequestType"
/>
```

3.1.4.1.2.2 FindMessageTrackingReportResponse Element

The [FindMessageTrackingReportResponse](#) element specifies the response to a [FindMessageTrackingReport](#) operation (section [3.1.4.1](#)) request.

```
<xs:element name="FindMessageTrackingReportResponse"
  type="m:FindMessageTrackingReportResponseMessageType"
/>
```

3.1.4.1.3 Messages

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

3.1.4.1.3.1 tns:FindMessageTrackingReportSoapIn Message

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

Part Name	Element/Type	Description
request	tns:FindMessageTrackingReport	Specifies the request.
RequestVersion	t:RequestServerVersion	Specifies the schema version for the

Part Name	Element/Type	Description
		FindMessageTrackingReport operation (section 3.1.4.1) request.

3.1.4.1.3.2 tns:FindMessageTrackingReportSoapOut Message

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

Part Name	Element/Type	Description
FindMessageTrackingReportResult	tns:FindMessageTrackingReportResponse	Specifies the response message.
ServerVersion	t:ServerVersionInfo	Specifies the schema version for the FindMessageTrackingReportSoapOut message.

3.1.4.2 GetMessageTrackingReport

The [GetMessageTrackingReport](#) operation gets tracking information about the specified messages.

```
<wsdl:operation name="GetMessageTrackingReport"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:GetMessageTrackingReportSoapIn" />
  <wsdl:output message="tns:GetMessageTrackingReportSoapOut" />
</wsdl:operation>
```

Request

Message Format	Description
tns:GetMessageTrackingReportSoapIn	Specifies the SOAP message that requests the report.

Response

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

3.1.4.2.1 Simple Types

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

3.1.4.2.1.1 t:MessageTrackingDeliveryStatusType Simple Type

The [MessageTrackingDeliveryStatusType](#) simple type specifies the status for message delivery.

```
<xs:simpleType name="MessageTrackingDeliveryStatusType">
  <xs:restriction
```



```

    base="xs:string"
  >
    <xs:enumeration
      value="Unsuccessful"
    />
    <xs:enumeration
      value="Pending"
    />
    <xs:enumeration
      value="Delivered"
    />
    <xs:enumeration
      value="Transferred"
    />
    <xs:enumeration
      value="Read"
    />
  </xs:restriction>
</xs:simpleType>

```

Enumeration

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

Value	Description
Unsuccessful	Specifies that the message was not delivered.
Pending	Specifies that the message is waiting for approval from a moderator.
Delivered	Specifies that the message was delivered to the recipients.
Transferred	Specifies that the message was transferred to a server outside the search scope.
Read	Specifies that the message was delivered and read by the recipients.

3.1.4.2.1.2 t:MessageTrackingEventDescriptionType Simple Type

The [MessageTrackingEventDescriptionType](#) element specifies the status of the message for an event in the tracking report.

```

<xs:simpleType name="MessageTrackingEventDescriptionType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Submitted"
    />
    <xs:enumeration
      value="Resolved"
    />
    <xs:enumeration
      value="Expanded"
    />
    <xs:enumeration

```

```
    value="Delivered"
  />
<xs:enumeration
  value="MovedToFolderByInboxRule"
 />
<xs:enumeration
  value="RulesCc"
 />
<xs:enumeration
  value="FailedGeneral"
 />
<xs:enumeration
  value="FailedModeration"
 />
<xs:enumeration
  value="FailedTransportRules"
 />
<xs:enumeration
  value="SmtplibSend"
 />
<xs:enumeration
  value="SmtplibSendCrossSite"
 />
<xs:enumeration
  value="SmtplibSendCrossForest"
 />
<xs:enumeration
  value="SmtplibReceive"
 />
<xs:enumeration
  value="Forwarded"
 />
<xs:enumeration
  value="Pending"
 />
<xs:enumeration
  value="PendingModeration"
 />
<xs:enumeration
  value="ApprovedModeration"
 />
<xs:enumeration
  value="QueueRetry"
 />
<xs:enumeration
  value="QueueRetryNoRetryTime"
 />
<xs:enumeration
  value="MessageDefer"
 />
<xs:enumeration
  value="TransferredToForeignOrg"
 />
<xs:enumeration
  value="TransferredToPartnerOrg"
 />
<xs:enumeration
  value="TransferredToLegacyExchangeServer"
 />
```

```

<xs:enumeration
  value="DelayedAfterTransferToPartnerOrg"
/>
<xs:enumeration
  value="Read"
/>
<xs:enumeration
  value="NotRead"
/>
</xs:restriction>
</xs:simpleType>

```

Enumeration

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

Value	Description
Submitted	Specifies that the message was submitted by the client.
Resolved	Specifies that the recipients for the message were successfully resolved.
Expanded	Specifies that a distribution list was expanded to deliver the message to the members.
Delivered	Specifies that the message was delivered to the recipient mailbox.
MovedToFolderByInboxRule	Specifies that the message was delivered to a different folder due to an Inbox rule.
RulesCc	Specifies that another recipient was copied on the message due to a transport rule.
FailedGeneral	Specifies that the message failed during delivery.
FailedModeration	Specifies that the delivery of the message failed because it was rejected by the moderator.
FailedTransportRules	Specifies that the delivery of the message failed due to a transport rule.
SmtptSend	Specifies that the message was sent over SMTP to the Internet and cannot be tracked further.
SmtptSendCrossSite	Specifies that the message was sent to a hub in a different site.
SmtptSendCrossForest	Specifies that the message was sent to a hub in a different trusted forest.
SmtptReceive	Specifies that the hub received a message over SMTP from a server on the Internet, or from a server that does not support tracking.
Forwarded	Specifies that the message was forwarded to another recipient.

Value	Description
Pending	Specifies that the message has not been delivered yet.
PendingModeration	Specifies that the message was sent to a moderator and is waiting for an approval message.
ApprovedModeration	Specifies that the message was approved by the moderator.
QueueRetry	Specifies that the message is in a transport queue and the retry time could not be retrieved.
QueueRetryNoRetryTime	Specifies that the message is in a transport queue and that the <EventData> XML element contains a string that represents the next time that the message will be retried.
MessageDefer	Specifies that the delivery of the message has been deferred for an unknown reason.
TransferredToForeignOrg	Specifies that the message was transferred to another organization or to a server in the organization that does not support tracking.
TransferredToPartnerOrg	Specifies that the message was transferred to a cross-premise organization.
TransferredToLegacyExchangeServer	Specifies that the message was transferred to an earlier version of the server, or to another server with a different schema.
DelayedAfterTransferToPartnerOrg	Specifies that the message delivery has been delayed during the transfer to a trusted organization.
Read	Specifies that the message is marked as read.
NotRead	Specifies that the message is marked as not read.

3.1.4.2.1.3 t:MessageTrackingReportTemplateType Simple Type

The [MessageTrackingReportTemplateType](#) simple type specifies the type of report to display.

```
<xs:simpleType name="MessageTrackingReportTemplateType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Summary"
    />
    <xs:enumeration
      value="RecipientPath"
    />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

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

Value	Description
Summary	Specifies that the report will display all the recipients of the message and the delivery status of the message to each recipient.
RecipientPath	Specifies that for a single recipient, the report will display a full history of the events that occurred.

3.1.4.2.2 Complex Types

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

3.1.4.2.2.1 m:GetMessageTrackingReportRequestType Complex Type

The **GetMessageTrackingReportRequestType** complex type specifies details for the type of report to retrieve. The **GetMessageTrackingReportRequestType** complex type extends the **BaseRequestType** complex type ([\[MS-OXWSCDATA\]](#) section 2.2.3.14).

```
<xs:complexType name="GetMessageTrackingReportRequestType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:all>
        <xs:element name="Scope"
          type="t:MessageTrackingScopeType"
        />
        <xs:element name="ReportTemplate"
          type="t:MessageTrackingReportTemplateType"
        />
        <xs:element name="RecipientFilter"
          type="t:EmailAddressType"
          minOccurs="0"
        />
        <xs:element name="MessageTrackingReportId"
          type="t:NonEmptyStringType"
        />
        <xs:element name="ReturnQueueEvents"
          type="xs:boolean"
          minOccurs="0"
        />
        <xs:element name="DiagnosticsLevel"
          type="xs:string"
          minOccurs="0"
        />
      </xs:all>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element	Type	Description
Scope	t:MessageTrackingScopeType	Specifies where to perform the search.
ReportTemplate	t:MessageTrackingReportTemplateType	Specifies the type of report to get.
RecipientFilter	t:EmailAddressType	Specifies a recipient address to use with the specified report.
MessageTrackingReportId	t:NonEmptyStringType	Specifies an identity string that was obtained from the FindMessageTrackingReport operation (section 3.1.4.1).
ReturnQueueEvents	xs:boolean	Specifies that the person who is running the task is in a privileged role.
DiagnosticsLevel	xs:string	Specifies timing and performance information that will be used to derive the report.

3.1.4.2.2.2 m:GetMessageTrackingReportResponseMessageType Complex Type

The **GetMessageTrackingReportResponseMessageType** complex type specifies the response for getting a report. The **GetMessageTrackingReportResponseMessageType** complex type extends the **ResponseMessageType** complex type ([\[MS-OXWSADATA\]](#) section 2.2.3.52).

```
<xs:complexType name="GetMessageTrackingReportResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="MessageTrackingReport"
          type="t:MessageTrackingReportType"
        />
        <xs:element name="Diagnostics"
          type="t:ArrayOfStringsType"
          minOccurs="0"
        />
        <xs:element name="Warnings"
          type="t:ArrayOfStringsType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element	Type	Description
MessageTrackingReport	t:MessageTrackingReportType	Specifies the tracking report that was requested.
Diagnostics	t:ArrayOfStringsType	Specifies timing and performance information that will be used to derive the report.
Warnings	t:ArrayOfStringsType	Specifies possible issues that will be used to derive the report.

3.1.4.2.2.3 t:ArrayOfRecipientTrackingEventType Complex Type

The [ArrayOfRecipientTrackingEventType](#) complex type specifies an array of one or more events for a message.

```
<xs:complexType name="ArrayOfRecipientTrackingEventType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="RecipientTrackingEvent"
      type="t:RecipientTrackingEventType"
    />
  </xs:choice>
</xs:complexType>
```

Child Elements

Element	Type	Description
RecipientTrackingEvent	t:RecipientTrackingEventType	Specifies an event for a message.

3.1.4.2.2.4 t:MessageTrackingReportType Complex Type

The [MessageTrackingReportType](#) complex type specifies the information to be included in the report.

```
<xs:complexType name="MessageTrackingReportType">
  <xs:all>
    <xs:element name="Sender"
      type="t:EmailAddressType"
      minOccurs="0"
    />
    <xs:element name="PurportedSender"
      type="t:EmailAddressType"
      maxOccurs="0"
    />
    <xs:element name="Subject"
      type="xs:string"
      minOccurs="0"
    />
    <xs:element name="SubmitTime"
      type="xs:dateTime"
    />
  </xs:all>
</xs:complexType>
```

```

        minOccurs="0"
      />
    <xs:element name="OriginalRecipients"
      type="t:ArrayOfEmailAddressesType"
      minOccurs="0"
    />
    <xs:element name="RecipientTrackingEvents"
      type="t:ArrayOfRecipientTrackingEventType"
    />
  </xs:all>
</xs:complexType>

```

Child Elements

Element	Type	Description
Sender	t:EmailAddressType	Specifies the e-mail address for the sender of the message.
PurportedSender	t:EmailAddressType	Specifies the e-mail address of the intended person sending the message.
Subject	xs:string	Specifies the subject of the message.
SubmitTime	xs:dateTime	Specifies the time at which the message was sent to the server.
OriginalRecipients	t:ArrayOfEmailAddressesType	Specifies the e-mail addresses of the message recipients.
RecipientTrackingEvents	t:ArrayOfRecipientTrackingEventType	Specifies the type of events to report.

3.1.4.2.2.5 t:RecipientTrackingEventType Complex Type

The [RecipientTrackingEventType](#) complex type specifies details for a specific event in the tracking report.

```

<xs:complexType name="RecipientTrackingEventType">
  <xs:all>
    <xs:element name="Date"
      type="xs:dateTime"
    />
    <xs:element name="Recipient"
      type="t:EmailAddressType"
    />
    <xs:element name="DeliveryStatus"
      type="t:MessageTrackingDeliveryStatusType"
    />
    <xs:element name="EventDescription"
      type="t:MessageTrackingEventDescriptionType"
    />
    <xs:element name="EventData"
      type="t:ArrayOfStringsType"
      minOccurs="0"
    />
    <xs:element name="Server "

```



```

        type="t:NonEmptyStringType"
    />
    <xs:element name="InternalId"
        type="xs:nonNegativeInteger"
    />
    <xs:element name="BccRecipient"
        type="xs:boolean"
        minOccurs="0"
    />
    <xs:element name="HiddenRecipient"
        type="xs:boolean"
        minOccurs="0"
    />
    <xs:element name="UniquePathId"
        type="t:NonEmptyStringType"
        minOccurs="0"
    />
</xs:all>
</xs:complexType>

```

Child Elements

Element	Type	Description
Date	xs:dateTime	Specifies the time at which the event occurred.
Recipient	t:EmailAddressType	Specifies the recipient for whom the event occurred.
DeliveryStatus	t:MessageTrackingDeliveryStatusType	Specifies the status for the message.
EventDescription	t:MessageTrackingEventDescriptionType	Specifies the processing step for the event.
EventData	t:ArrayOfStringsType	Specifies data that is associated with the processing step for the event.
Server	t:NonEmptyStringType	Specifies the physical server where the event occurred.
InternalId	xs:nonNegativeInteger	Specifies an integer value for the event.
BccRecipient	xs:boolean	Specifies that the recipient was addressed as a blind carbon copy (BCC) recipient .
HiddenRecipient	xs:boolean	Specifies that the recipient was added by an organization policy that should be hidden from unprivileged users.
UniquePathId	t:NonEmptyStringType	Specifies a string that is different for each path.

3.1.4.2.3 Elements

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

3.1.4.2.3.1 GetMessageTrackingReport Element

The [GetMessageTrackingReport](#) element specifies the request to get a tracking report.

```
<xs:element name="GetMessageTrackingReport"
  type="m:GetMessageTrackingReportRequestType"
/>
```

3.1.4.2.3.2 GetMessageTrackingReportResponse Element

The [GetMessageTrackingReportResponse](#) element specifies the response for getting a report.

```
<xs:element name="GetMessageTrackingReportResponse"
  type="m:GetMessageTrackingReportResponseMessageType"
/>
```

3.1.4.2.4 Messages

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

3.1.4.2.4.1 tns:GetMessageTrackingReportSoapIn Message

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

Part Name	Element/Type	Description
request	tns:GetMessageTrackingReport	Specifies the request.
RequestVersion	t:RequestServerVersion	Specifies the schema version for the GetMessageTrackingReport operation (section 3.1.4.2) request.

3.1.4.2.4.2 tns:GetMessageTrackingReportSoapOut Message

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

Part Name	Element/Type	Description
GetMessageTrackingReportResult	tns:GetMessageTrackingReportResponse	Specifies the response message.
ServerVersion	t:ServerVersionInfo	Specifies the schema version for the GetMessageTrackingReportSoapOut 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 Message Tracking Web Service protocol does not use any additional security mechanisms.

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 that is specified in this document. The contents of each file are included in this section.

File name	Description	Section
MS-OXWSMTRK.wsdl	Contains the WSDL for the implementation of this protocol.	WSDL 6.1
MS-OXWSMTRK-types.xsd	Contains the XML schema type definitions that are used in this protocol.	Types Schema 6.2
MS-OXWSMTRK-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	Messages Schema 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-OXWSMTRK-types.xsd or MS-OXWSMTRK-messages.xsd schemas have to be placed in the common folder along with the files.

6.1 WSDL

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

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

```

        <wsdl:input message="tns:GetMessageTrackingReportSoapIn" />
        <wsdl:output message="tns:GetMessageTrackingReportSoapOut" />
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
<wsdl:documentation>
<wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/" />
</wsdl:documentation>
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="FindMessageTrackingReport">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindMessageTrackingReport" />
        <wsdl:input>
            <soap:body parts="request" use="literal" />
            <soap:header message="tns:FindMessageTrackingReportSoapIn"
part="RequestVersion" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="FindMessageTrackingReportResult" use="literal" />
            <soap:header message="tns:FindMessageTrackingReportSoapOut"
part="ServerVersion" use="literal" />
        </wsdl:output>
    </wsdl:operation>

    <wsdl:operation name="GetMessageTrackingReport">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetMessageTrackingReport" />
        <wsdl:input>
            <soap:body parts="request" use="literal" />
            <soap:header message="tns:GetMessageTrackingReportSoapIn"
part="RequestVersion" use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="GetMessageTrackingReportResult" use="literal" />
            <soap:header message="tns:GetMessageTrackingReportSoapOut"
part="ServerVersion" use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:message name="FindMessageTrackingReportSoapIn">
    <wsdl:part name="request" element="tns:FindMessageTrackingReport" />
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
</wsdl:message>
<wsdl:message name="FindMessageTrackingReportSoapOut">
    <wsdl:part name="FindMessageTrackingReportResult"
element="tns:FindMessageTrackingReportResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
</wsdl:message>

<wsdl:message name="GetMessageTrackingReportSoapIn">
    <wsdl:part name="request" element="tns:GetMessageTrackingReport" />
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
</wsdl:message>
<wsdl:message name="GetMessageTrackingReportSoapOut">
    <wsdl:part name="GetMessageTrackingReportResult"
element="tns:GetMessageTrackingReportResponse" />
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
</wsdl:message>

```

```

</wsdl:message>
</wsdl:definitions>

```

6.2 Types Schema

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

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

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

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:complexType name="ArrayOfFindMessageTrackingSearchResultType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="MessageTrackingSearchResult"
type="t:FindMessageTrackingSearchResultType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="FindMessageTrackingSearchResultType">
    <xs:all>
      <xs:element name="Subject" type="xs:string"/>
      <xs:element name="Sender" type="t:EmailAddressType"/>
      <xs:element name="Recipients" type="t:ArrayOfRecipientsType"/>
      <xs:element name="SubmittedTime" type="xs:dateTime"/>
      <xs:element name="MessageTrackingReportId" type="t:NonEmptyStringType"/>
      <xs:element name="PreviousHopServer" type="t:NonEmptyStringType"
minOccurs="0"/>
    </xs:all>
  </xs:complexType>
  <xs:simpleType name="MessageTrackingDeliveryStatusType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Unsuccessful"/>
      <xs:enumeration value="Pending"/>
      <xs:enumeration value="Delivered"/>
      <xs:enumeration value="Transferred"/>
      <xs:enumeration value="Read"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="MessageTrackingEventDescriptionType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Submitted"/>
      <xs:enumeration value="Resolved"/>
      <xs:enumeration value="Expanded"/>
      <xs:enumeration value="Delivered"/>
    </xs:restriction>
  </xs:simpleType>

```



```

        <xs:enumeration value="MovedToFolderByInboxRule" />
        <xs:enumeration value="RulesCc" />
        <xs:enumeration value="FailedGeneral" />
        <xs:enumeration value="FailedModeration" />
        <xs:enumeration value="FailedTransportRules" />
        <xs:enumeration value="SmtptSend" />
        <xs:enumeration value="SmtptSendCrossSite" />
        <xs:enumeration value="SmtptSendCrossForest" />
        <xs:enumeration value="SmtptReceive" />
        <xs:enumeration value="Forwarded" />
        <xs:enumeration value="Pending" />
        <xs:enumeration value="PendingModeration" />
        <xs:enumeration value="ApprovedModeration" />
        <xs:enumeration value="QueueRetry" />
        <xs:enumeration value="QueueRetryNoRetryTime" />
        <xs:enumeration value="MessageDefer" />
        <xs:enumeration value="TransferredToForeignOrg" />
        <xs:enumeration value="TransferredToPartnerOrg" />
        <xs:enumeration value="TransferredToLegacyExchangeServer" />
        <xs:enumeration value="DelayedAfterTransferToPartnerOrg" />
        <xs:enumeration value="Read" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MessageTrackingScopeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Organization" />
        <xs:enumeration value="Forest" />
        <xs:enumeration value="Site" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MessageTrackingReportTemplateType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Summary" />
        <xs:enumeration value="RecipientPath" />
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="RecipientTrackingEventType">
    <xs:all>
        <xs:element name="Date" type="xs:dateTime" />
        <xs:element name="Recipient" type="t:EmailAddressType" />
        <xs:element name="DeliveryStatus" type="t:MessageTrackingDeliveryStatusType" />
        <xs:element name="EventDescription"
type="t:MessageTrackingEventDescriptionType" />
        <xs:element name="EventData" type="t:ArrayOfStringsType" minOccurs="0" />
        <xs:element name="Server" type="t:NonEmptyStringType" />
        <xs:element name="InternalId" type="xs:nonNegativeInteger" />
        <xs:element name="BccRecipient" type="xs:boolean" minOccurs="0" />
        <xs:element name="HiddenRecipient" type="xs:boolean" minOccurs="0" />
        <xs:element name="UniquePathId" type="t:NonEmptyStringType" minOccurs="0" />
    </xs:all>
</xs:complexType>
<xs:complexType name="MessageTrackingReportType">
    <xs:all>
        <xs:element name="Sender" type="t:EmailAddressType" minOccurs="0" />
        <xs:element name="Subject" type="xs:string" minOccurs="0" />
        <xs:element name="SubmitTime" type="xs:dateTime" minOccurs="0" />
        <xs:element name="OriginalRecipients" type="t:ArrayOfEmailAddressesType"
minOccurs="0" />
    </xs:all>

```

```

        <xs:element name="RecipientTrackingEvents"
type="t:ArrayOfRecipientTrackingEventType"/>
    </xs:all>
</xs:complexType>
<xs:complexType name="ArrayOfRecipientTrackingEventType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="RecipientTrackingEvent"
type="t:RecipientTrackingEventType"/>
    </xs:choice>
</xs:complexType>
</xs:schema>

```

6.3 Messages Schema

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

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

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

```

<?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:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
    <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSMTRK-types.xsd"/>
    <xs:complexType name="FindMessageTrackingReportRequestType">
        <xs:complexContent>
            <xs:extension base="m:BaseRequestType">
                <xs:all>
                    <xs:element name="Scope" type="t:MessageTrackingScopeType"/>
                    <xs:element name="Domain" type="t:NonEmptyStringType"/>
                    <xs:element name="Sender" type="t:EmailAddressType" minOccurs="0"/>
                    <xs:element name="Recipient" type="t:EmailAddressType"
minOccurs="0"/>
                    <xs:element name="Subject" type="xs:string" minOccurs="0"/>
                    <xs:element name="StartDateTime" type="xs:dateTime" minOccurs="0"/>
                    <xs:element name="EndDateTime" type="xs:dateTime" minOccurs="0"/>
                    <xs:element name="MessageId" type="t:NonEmptyStringType"
minOccurs="0"/>
                    <xs:element name="FederatedDeliveryMailbox"
type="t:EmailAddressType" minOccurs="0"/>

```

```

        <xs:element name="DiagnosticsLevel" type="xs:string" minOccurs="0"/>
    </xs:all>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="FindMessageTrackingReport"
type="m:FindMessageTrackingReportRequestType"/>
<xs:complexType name="FindMessageTrackingReportResponseMessageType">
    <xs:complexContent>
        <xs:extension base="m:ResponseMessageType">
            <xs:sequence>
                <xs:element name="Diagnostics" type="t:ArrayOfStringsType"
minOccurs="0"/>
                <xs:element name="MessageTrackingSearchResults"
type="t:ArrayOfFindMessageTrackingSearchResultType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="FindMessageTrackingReportResponse"
type="m:FindMessageTrackingReportResponseMessageType"/>
<xs:complexType name="GetMessageTrackingReportRequestType">
    <xs:complexContent>
        <xs:extension base="m:BaseRequestType">
            <xs:all>
                <xs:element name="Scope" type="t:MessageTrackingScopeType"/>
                <xs:element name="ReportTemplate"
type="t:MessageTrackingReportTemplateType"/>
                <xs:element name="RecipientFilter" type="t:EmailAddressType"
minOccurs="0"/>
                <xs:element name="MessageTrackingReportId"
type="t:NonEmptyStringType"/>
                <xs:element name="ReturnQueueEvents" type="xs:boolean"
minOccurs="0"/>
                <xs:element name="DiagnosticsLevel" type="xs:string" minOccurs="0"/>
            </xs:all>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="GetMessageTrackingReport"
type="m:GetMessageTrackingReportRequestType"/>
<xs:complexType name="GetMessageTrackingReportResponseMessageType">
    <xs:complexContent>
        <xs:extension base="m:ResponseMessageType">
            <xs:sequence>
                <xs:element name="MessageTrackingReport"
type="t:MessageTrackingReportType"/>
                <xs:element name="Diagnostics" type="t:ArrayOfStringsType"
minOccurs="0"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="GetMessageTrackingReportResponse"
type="m:GetMessageTrackingReportResponseMessageType"/>
</xs:schema>

```

7 Appendix B: Product Behavior

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

- Microsoft Exchange Server 2010

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

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

8 Change Tracking

This section identifies changes made to [MS-OXWSMTRK] protocol documentation between November 2009 and February 2010 releases. Changes are classed as major, minor, or editorial.

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.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

- New content added.
- Content update.
- 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 always have the revision 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.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
3.1.4.1.3 Messages	Changed "XML schema" to "WSDL message".	N	Content update.

9 Index

A

[Applicability](#) 7

C

[Capability negotiation](#) 7

[Change tracking](#) 37

Client

[abstract data model](#) 27

initialization ([section 3.2.2](#) 27, [section 3.2.3](#) 27)

[local events](#) 27

[message processing](#) 27

[overview](#) 27

[sequencing rules](#) 27

[timer events](#) 27

F

[Full WSDL](#) 30

G

[Glossary](#) 5

I

[Introduction](#) 5

M

Messages

[overview](#) 8

[syntax](#) 8

[transport](#) 8

O

[Overview](#) 6

P

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 36

[Protocol details](#) 10

R

[References](#) 5

[informative](#) 6

[normative](#) 5

[Relationship to other protocols](#) 6

S

Security

[implementer considerations](#) 29

[overview](#) 29

[parameter index](#) 29

Server

[abstract data model](#) 10

[initialization](#) 10

[local events](#) 27

[message processing](#) 10

[sequencing rules](#) 10

[timer events](#) 26

[timers](#) 10

Sever

[overview](#) 10

[Standards assignments](#) 7

T

[Tracking changes](#) 37

V

[Vendor-extensible fields](#) 7

[Versioning](#) 7