

[MS-OXWSRSLNM]:

Resolve Recipient Names Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

Preliminary

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0	Major	Initial Availability.
11/4/2009	1.1.0	Minor	Updated the technical content.
2/10/2010	1.2.0	Minor	Updated the technical content.
5/5/2010	1.2.1	Editorial	Revised and edited the technical content.
8/4/2010	1.3	Minor	Clarified the meaning of the technical content.
11/3/2010	2.0	Major	Significantly changed the technical content.
3/18/2011	2.1	Minor	Clarified the meaning of the technical content.
8/5/2011	3.0	Major	Significantly changed the technical content.
10/7/2011	4.0	Major	Significantly changed the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	5.1	Minor	Clarified the meaning of the technical content.
10/8/2012	5.2	Minor	Clarified the meaning of the technical content.
2/11/2013	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	5.3	Minor	Clarified the meaning of the technical content.
7/31/2014	5.3	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	5.4	Minor	Clarified the meaning of the technical content.
5/26/2015	6.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments	10
2	Messages	11
2.1	Transport	11
2.2	Common Message Syntax	11
2.2.1	Namespaces	11
2.2.2	Messages	11
2.2.3	Elements	11
2.2.4	Complex Types	12
2.2.5	Simple Types	12
2.2.6	Attributes	12
2.2.7	Groups	12
2.2.8	Attribute Groups	12
3	Protocol Details	13
3.1	ExchangeServicePortType Server Details	13
3.1.1	Server Abstract Data Model	13
3.1.2	Server Timers	13
3.1.3	Server Initialization	13
3.1.4	Message Processing Events and Sequencing Rules	13
3.1.4.1	ResolveNames Operation	13
3.1.4.1.1	Messages	14
3.1.4.1.1.1	tns:ResolveNamesSoapIn	14
3.1.4.1.1.2	tns:ResolveNamesSoapOut	15
3.1.4.1.2	Elements	15
3.1.4.1.2.1	ResolveNames Element	15
3.1.4.1.2.2	ResolveNamesResponse Element	16
3.1.4.1.3	Complex Types	16
3.1.4.1.3.1	t:ArrayOfResolutionType Complex Type	16
3.1.4.1.3.2	t:Resolutiontype Complex Type	17
3.1.4.1.3.3	m:ResolveNamesResponseMessageType Complex Type	17
3.1.4.1.3.4	m:ResolveNamesResponseType Complex Type	18
3.1.4.1.3.5	m:ResolveNamesType Complex Type	18
3.1.4.1.4	Simple Types	19
3.1.4.1.4.1	t:ResolveNamesSearchScopeType Simple Type	19
3.1.5	Timer Events	20
3.1.6	Other Local Events	20
4	Protocol Examples	21
5	Security	22
5.1	Security Considerations for Implementers	22
5.2	Index of Security Parameters	22

6	Appendix A: Full WSDL	23
7	Appendix B: Full XML Schema.....	25
7.1	Messages Schema	25
7.2	Types Schema	26
8	Appendix C: Product Behavior	27
9	Change Tracking.....	28
10	Index.....	30

Preliminary

1 Introduction

The Resolve Recipient Names Web Service Protocol enables a client that has incomplete **recipient** identifying information to retrieve a list of matching and similar recipients that are known to the server.

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

1.1 Glossary

The following terms are specific to this document:

contact: A person, company, or other entity that is stored in a directory and is associated with one or more unique identifiers and attributes (2), such as an Internet message address or login name.

Contacts folder: A Folder object that contains Contact objects.

directory service (DS): A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also Active Directory.

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

email address: A string that identifies a user and enables the user to receive Internet messages.

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

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

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

mailbox: A message store that contains email, calendar items, and other Message objects for a single recipient.

recipient: An entity that is in an address list, can receive email messages, and contains a set of attributes (1). Each attribute has a set of associated values.

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

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

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

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

SOAP message: An **XML** document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [SOAP1.2-1/2007] section 5 for more information.

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

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

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

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

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

XML: The Extensible Markup Language, as described in [\[XML1.0\]](#).

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

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

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

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

1.2 References

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

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

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

[MS-OXWSCONT] Microsoft Corporation, "[Contacts Web Service Protocol](#)".

[MS-OXWSFOLD] Microsoft Corporation, "[Folders and Folder Permissions Web Service Protocol](#)".

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

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

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

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

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

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

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

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

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

1.2.2 Informative References

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

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

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

1.3 Overview

The Resolve Recipient Names Web Service Protocol enables a client to retrieve a list of **email addresses** or **contacts** that are known to the server. The client can provide the full or partial name for the intended recipient. The server that is implementing the Resolve Recipient Names Web Service Protocol will return a list of matching or similarly named recipients. The client can then use those names when addressing items for the server to send. The server can return valid recipients that are known to it; for example, recipient names that are stored in a **directory service** or in a global or user-specific list of contacts.

1.4 Relationship to Other Protocols

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

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

The Resolve Recipient Names Web Service Protocol uses SOAP over **HTTP**, as specified in [\[RFC2616\]](#), and SOAP over **HTTPS**, as described in [\[RFC2818\]](#), as shown in the layering diagram.

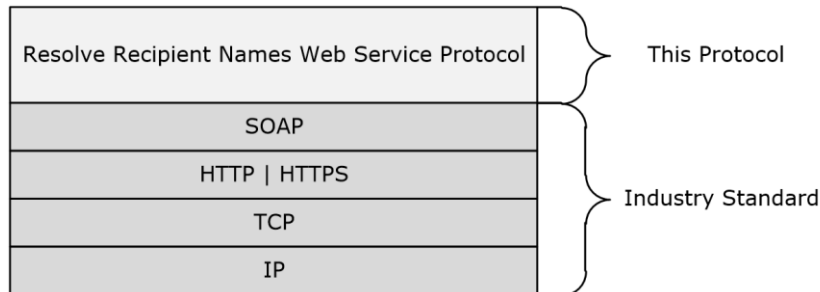


Figure 1: This protocol in relation to other protocols

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

1.5 Prerequisites/Preconditions

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

1.6 Applicability Statement

The Resolve Recipient Names Web Service Protocol is applicable to clients that obtain a list of candidate recipients that are possible matches for ambiguous or partial names that are provided to the client. The server returns the list of candidate matches, and the client application determines whether any of the listed candidates is the intended recipient.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple transports with SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.11, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Localization:** This protocol includes text strings in various messages. Localization considerations for such strings are specified in section [3.1.4](#).

- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Preliminary

2 Messages

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

2.1 Transport

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

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

2.2 Common Message Syntax

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

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific **XML namespace prefix** with each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and is 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	
m	http://schemas.microsoft.com/exchange/services/2006/messages	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] [XMLSCHEMA2]

2.2.2 Messages

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

2.2.3 Elements

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

2.2.4 Complex Types

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

2.2.5 Simple Types

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

2.2.6 Attributes

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

2.2.7 Groups

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

2.2.8 Attribute Groups

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

Preliminary

3 Protocol Details

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

3.1 ExchangeServicePortType Server Details

The Resolve Recipient Names Web Service Protocol defines a single port type with one operation.

3.1.1 Server Abstract Data Model

None.

3.1.2 Server Timers

None.

3.1.3 Server Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

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

Operation name	Description
ResolveNames	Returns a list of candidate recipient names that are known to the server and that match or are similar to the recipient name provided in the request by the client.

3.1.4.1 ResolveNames Operation

The **ResolveNames** operation returns a list of candidate recipients that match an ambiguous recipient name that is provided by the client.

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

```
<wsdl:operation name="ResolveNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:input message="tns:ResolveNamesSoapIn"/>
  <wsdl:output message="tns:ResolveNamesSoapOut"/>
</wsdl:operation>
```

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

```
<wsdl:operation name="ResolveNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <soap:operation
    soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ResolveNames"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
```

```

<wsdl:input>
  <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <soap:header message="tns:ResolveNamesSoapIn" part="Impersonation" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <soap:header message="tns:ResolveNamesSoapIn" part="MailboxCulture" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <soap:header message="tns:ResolveNamesSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
</wsdl:input>
<wsdl:output>
  <soap:body parts="ResolveNamesResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <soap:header message="tns:ResolveNamesSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
</wsdl:output>
</wsdl:operation>

```

3.1.4.1.1 Messages

The following table lists and describes the WSDL messages that are specific to the **ResolveNames** operation.

Message name	Description
ResolveNamesSoapIn	Specifies the SOAP message that requests name resolution from the server.
ResolveNamesSoapOut	Specifies the SOAP message that is returned in response.

3.1.4.1.1.1 tns:ResolveNamesSoapIn

The **ResolveNamesSoapIn** WSDL message specifies the **ResolveNames** operation request to match an ambiguous recipient name.

```

<wsdl:message name="ResolveNamesSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="request" element="tns:ResolveNames"/>
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>

```

The **ResolveNamesSoapIn** WSDL message is the input message for the **SOAP action** <http://schemas.microsoft.com/exchange/services/2006/messages/ResolveNames>.

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

Part name	Element/type	Description
request	tns:ResolveNames (section 3.1.4.1.2.1)	Specifies the SOAP body of the request.
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)	Specifies the identifier of the account to impersonate.
MailboxCulture	t:MailboxCulture ([MS-OXWSCDATA])	Specifies a SOAP header that identifies the culture to be used for accessing the server. The

Part name	Element/type	Description
	section 2.2.3.7)	cultures are defined by [RFC3066] .
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the request.

3.1.4.1.1.2 tns:ResolveNamesSoapOut

The **ResolveNamesSoapOut** WSDL message specifies the server response to the **ResolveNames** operation request to match an ambiguous recipient name.

```
<wsdl:message name="ResolveNamesSoapOut"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:part name="ResolveNamesResult" element="tns:ResolveNamesResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **ResolveNamesSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/ResolveNames>.

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

Part name	Element/type	Description
ResolveNamesResult	tns:ResolveNamesResponse (section 3.1.4.1.2.2)	Specifies the SOAP body of the response.
ServerVersion	t:ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies a SOAP header that identifies the server version for the response.

3.1.4.1.2 Elements

The following table lists and describes the XML schema elements that are specific to the **ResolveNames** operation.

Element name	Description
ResolveNames	Specifies the parameters and the unresolved recipient name to the server.
ResolveNamesResponse	Specifies the candidate matches that are returned in the server response.

3.1.4.1.2.1 ResolveNames Element

The **ResolveNames** element is a container that specifies the parameters and the unresolved recipient name to the server.

```
<xs:element name="ResolveNames"
type="m:ResolveNamesType"
/>
```

3.1.4.1.2.2 ResolveNamesResponse Element

The **ResolveNamesResponse** element is a container that specifies the candidate matches that are returned in the server response.

```
<xs:element name="ResolveNamesResponse"
  type="m:ResolveNamesResponseType"
/>
```

3.1.4.1.3 Complex Types

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

Complex type name	Description
ArrayOfResolutionType	Specifies the list of matching recipient candidates that are returned by the server.
ResolutionType	Specifies the type of a matching recipient candidate that is returned by the server.
ResolveNamesResponseMessageType	Specifies the status and result of a ResolveNames operation.
ResolveNamesResponseType	Specifies the contents of the response from the server.
ResolveNamesType	Specifies the contents of a request from the client to locate matching recipient candidates.

3.1.4.1.3.1 t:ArrayOfResolutionType Complex Type

The **ArrayOfResolutionType** complex type specifies the list of matching recipient candidates that are returned by the server. Note that the server returns a maximum of 100 candidate matches.

```
<xs:complexType name="ArrayOfResolutionType">
  <xs:sequence>
    <xs:element name="Resolution"
      type="t:ResolutionType"
      maxOccurs="100"
      minOccurs="0"
    />
  </xs:sequence>
  <xs:attributeGroup
    ref="t:FindResponsePagingAttributes"
  />
</xs:complexType>
```

The following table lists and describes the child element of the **ArrayOfResolutionType** complex type.

Element name	Type	Description
Resolution	t:ResolutionType (section 3.1.4.1.3.2)	Specifies a single resolved entity.

The attribute group of the **ArrayOfResolutionType** complex type is the **FindResponsePagingAttributes** attribute group, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.8.1.

3.1.4.1.3.2 t:Resolutiontype Complex Type

The **ResolutionType** complex type specifies the type of matching recipient candidate that is returned by the server.

```
<xs:complexType name="ResolutionType">
  <xs:sequence>
    <xs:element name="Mailbox"
      type="t:EmailAddressType"
    />
    <xs:element name="Contact"
      type="t:ContactItemType"
      minOccurs="0"
    />
  </xs:sequence>
</xs:complexType>
```

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

Element name	Type	Description
Mailbox	t:EmailAddressType ([MS-OXWSCDATA] section 2.2.4.30)	Specifies an email address.
Contact	t:ContactItemType ([MS-OXWSCONT] section 2.2.4.3)	Specifies a contact item.

3.1.4.1.3.3 m:ResolveNamesResponseMessageType Complex Type

The **ResolveNamesResponseMessageType** complex type specifies the status and result of a **ResolveNames** operation. The **ResolveNamesResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```
<xs:complexType name="ResolveNamesResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    />
    <xs:sequence>
      <xs:element name="ResolutionSet"
        type="t:ArrayOfResolutionType"
        minOccurs="0"
      />
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>
```

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

Element name	Type	Description
ResolutionSet	t:ArrayOfResolutionType (section 3.1.4.1.3.1)	Specifies the list of matching recipient candidates that are returned by the server.

3.1.4.1.3.4 m:ResolveNamesResponseType Complex Type

The **ResolveNamesResponseType** complex type specifies the contents of the response from the server. The **ResolveNamesResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

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

3.1.4.1.3.5 m:ResolveNamesType Complex Type

The **ResolveNamesType** complex type specifies the contents of a request from the client to locate matching recipient candidates. The **ResolveNamesType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```
<xs:complexType name="ResolveNamesType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
    <xs:sequence>
      <xs:element name="ParentFolderIds"
        type="t:NonEmptyArrayOfBaseFolderIdsType"
        minOccurs="0"
      />
      <xs:element name="UnresolvedEntry"
        type="t:NonEmptyStringType"
      />
    </xs:sequence>
    <xs:attribute name="ReturnFullContactData"
      type="xs:boolean"
      use="required"
    />
    <xs:attribute name="SearchScope"
      type="t:ResolveNamesSearchScopeType"
      default="ActiveDirectoryContacts"
    />
    <xs:attribute name="ContactDataShape"
      type="t:DefaultShapeNamesType"
    />
  </xs:extension>
</xs:complexType>
```

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

Element name	Type	Description
ParentFolderIds	t:NonEmptyArrayOfBaseFolderIdsType ([MS-OXWSFOLD] section 3.1.4.6.3.3)	Specifies the ID of the parent folder that contains the item.
UnresolvedEntry	t:NonEmptyStringType ([MS-OXWSCDATA] section 2.2.5.20)	Specifies the name of a contact or distribution list to be resolved.

The following table lists and describes the attributes of the **ResolveNamesType** complex type.

Attribute name	Type	Description
ReturnFullContactData	xs:boolean [XMLSCHEMA2]	Specifies whether all the data is to be returned for a resolved contact.
SearchScope	t:ResolveNamesSearchScopeType (section 3.1.4.1.4.1)	Specifies a location where the server searches for candidate matches.
ContactDataShape	t:DefaultShapeNamesType ([MS-OXWSCDATA] section 2.2.5.7)	Specifies the standard set of properties to be returned for the resolved names. This attribute SHOULD <1> be included.

3.1.4.1.4 Simple Types

The following table lists and describes the XML schema simple type that is specific to the **ResolveNames** operation.

Simple type name	Description
ResolveNamesSearchScopeType	Specifies a location where the server searches for match candidates.

3.1.4.1.4.1 t:ResolveNamesSearchScopeType Simple Type

The **ResolveNamesSearchScopeType** simple type specifies a location where the server searches for match candidates.

```
<xs:simpleType name="ResolveNamesSearchScopeType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="ActiveDirectory"
    />
    <xs:enumeration
      value="ActiveDirectoryContacts"
    />
    <xs:enumeration
      value="Contacts"
    />
  </xs:restriction>
</xs:simpleType>
```

```

    <xs:enumeration
      value="ContactsActiveDirectory"
    />
  </xs:restriction>
</xs:simpleType>

```

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

Value name	Description
ActiveDirectory	Specifies that the server searches for match candidates in the directory service.
ActiveDirectoryContacts	Specifies that the server searches for match candidates in the directory service first, then in the Contacts folder of the mailbox .
Contacts	Specifies that the server searches for match candidates in the Contacts folder of the mailbox.
ContactsActiveDirectory	Specifies that the server searches for match candidates in the Contacts folder of the mailbox first, then in the directory service.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

The following example of a **ResolveNames** operation shows how to resolve the entry **User2**.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <soap:Body>
    <ResolveNames xmlns="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
        ReturnFullContactData="true">
      <UnresolvedEntry>User2</UnresolvedEntry>
    </ResolveNames>
  </soap:Body>
</soap:Envelope>
```

The following example shows a successful response to a **ResolveNames** operation.

```
<?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:Header>
    <t:ServerVersionInfo MajorVersion="8" MinorVersion="0" MajorBuildNumber="685"
      MinorBuildNumber="8"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types" />
  </soap:Header>
  <soap:Body>
    <ResolveNamesResponse
      xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <m:ResponseMessages>
        <m:ResolveNamesResponseMessage ResponseClass="Success">
          <m:ResponseCode>NoError</m:ResponseCode>
          <m:ResolutionSet TotalItemsInView="1" IncludesLastItemInRange="true">
            <t:Resolution>
              <t:Mailbox>
                <t:Name>User2</t:Name>
                <t:EmailAddress>User2@example.com</t:EmailAddress>
                <t:RoutingType>SMTP</t:RoutingType>
                <t:MailboxType>Mailbox</t:MailboxType>
              </t:Mailbox>
              <t:Contact>
                <t:DisplayName>User2</t:DisplayName>
                <t:EmailAddresses>
                  <t:Entry Key="EmailAddress1">SMTP:User2@example.com</t:Entry>
                </t:EmailAddresses>
                <t:ContactSource>ActiveDirectory</t:ContactSource>
              </t:Contact>
            </t:Resolution>
          </m:ResolutionSet>
        </m:ResolveNamesResponseMessage>
      </m:ResponseMessages>
    </ResolveNamesResponse>
  </soap:Body>
</soap:Envelope>
```

5 Security

5.1 Security Considerations for Implementers

The Resolve Recipient Names Web Service Protocol does not use any additional security mechanisms.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: Full WSDL

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

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

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSRSLNM-types.xsd or MS-OXWSRSLNM-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

This section contains the contents of the MS-OXWSRSLNM.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"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
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">
      <xs:include schemaLocation="MS-OXWSRSLNM-messages.xsd"/>
    </xs:schema>
    <xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="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:include schemaLocation="MS-OXWSRSLNM-types.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ExchangeServicePortType">
    <wsdl:operation name="ResolveNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <wsdl:input message="tns:ResolveNamesSoapIn"/>
      <wsdl:output message="tns:ResolveNamesSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:message name="ResolveNamesSoapIn" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="request" element="tns:ResolveNames"/>
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
  </wsdl:message>
  <wsdl:message name="ResolveNamesSoapOut" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:part name="ResolveNamesResult" element="tns:ResolveNamesResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
  </wsdl:message>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
```

```
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/" />
</wsdl:documentation>
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  <wsdl:operation name="ResolveNames" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ResolveNames"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
      <wsdl:input>
        <soap:body parts="request" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
          <soap:header message="tns:ResolveNamesSoapIn" part="Impersonation" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
          <soap:header message="tns:ResolveNamesSoapIn" part="MailboxCulture" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
          <soap:header message="tns:ResolveNamesSoapIn" part="RequestVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
        </wsdl:input>
        <wsdl:output>
          <soap:body parts="ResolveNamesResult" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
          <soap:header message="tns:ResolveNamesSoapOut" part="ServerVersion" use="literal"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```

Preliminary

7 Appendix B: Full XML Schema

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

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

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSRSLNM-types.xsd or MS-OXWSRSLNM-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

7.1 Messages Schema

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

MS-OXWSRSLNM-messages.xsd includes or imports the files listed in the following table. For the schema file to operate correctly, these files have to be in the folder that contains the WSDL, types schema, and messages schema files for this protocol.

File name	Defining specification
MS-OXWSCDATA-messages.xsd	[MS-OXWSCDATA] section 7.1
MS-OXWSRSLNM-types.xsd	section 7.2

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages"
  elementFormDefault="qualified" version="Exchange2016" id="messages">
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
    schemaLocation="MS-OXWSRSLNM-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:complexType name="ResolveNamesResponseMessageType">
    <xs:complexContent>
      <xs:extension base="m:ResponseMessageType">
        <xs:sequence>
          <xs:element name="ResolutionSet" type="t:ArrayOfResolutionType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ResolveNamesResponseType">
    <xs:complexContent>
      <xs:extension base="m:BaseResponseMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="ResolveNamesType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:element name="ParentFolderIds" type="t:NonEmptyArrayOfBaseFolderIdsType"
            minOccurs="0"/>
          <xs:element name="UnresolvedEntry" type="t:NonEmptyStringType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

```

    <xs:attribute name="ReturnFullContactData" type="xs:boolean" use="required"/>
    <xs:attribute name="SearchScope" type="t:ResolveNamesSearchScopeType"
default="ActiveDirectoryContacts"/>
    <xs:attribute name="ContactDataShape" type="t:DefaultShapeNamesType"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="ResolveNames" type="m:ResolveNamesType"/>
<xs:element name="ResolveNamesResponse" type="m:ResolveNamesResponseType"/>
</xs:schema>

```

7.2 Types Schema

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

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

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2

```

<?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="Exchange2016" id="types">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:complexType name="ArrayOfResolutionType">
    <xs:sequence>
      <xs:element name="Resolution" type="t:ResolutionType" minOccurs="0" maxOccurs="100"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResolutionType">
    <xs:sequence>
      <xs:element name="Mailbox" type="t:EmailAddressType"/>
      <xs:element name="Contact" type="t:ContactItemType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ResolveNamesSearchScopeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ActiveDirectory"/>
      <xs:enumeration value="ActiveDirectoryContacts"/>
      <xs:enumeration value="Contacts"/>
      <xs:enumeration value="ContactsActiveDirectory"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>

```

8 Appendix C: Product Behavior

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

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

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

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

[<1> Section 3.1.4.1.3.5](#): Exchange 2007, Exchange 2010 and Microsoft Exchange Server 2010 Service Pack 1 (SP1) do not support the **ContactDataShape** attribute. The **ContactDataShape** attribute was introduced in Microsoft Exchange Server 2010 Service Pack 2 (SP2).

9 Change Tracking

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
8 Appendix C: Product Behavior	Added Exchange 16 to list of supported products.	Y	Content update.

Preliminary

10 Index

A

Abstract data model
[server](#) 13
[Applicability](#) 9
[Attribute groups](#) 12
[Attributes](#) 12

C

[Capability negotiation](#) 9
[Change tracking](#) 28
[Complex types](#) 12

D

Data model - abstract
[server](#) 13

E

Events
[local - server](#) 20
[timer - server](#) 20

Examples

[ResolveNames operation](#) 21

F

[Fields - vendor-extensible](#) 10
[Full WSDL](#) 23
[Full XML schema](#) 25
[Messages Schema](#) 25
[Types Schema](#) 26

G

[Glossary](#) 6
[Groups](#) 12

I

[Implementer - security considerations](#) 22
[Index of security parameters](#) 22
[Informative references](#) 8
Initialization
[server](#) 13
[Introduction](#) 6

L

Local events
[server](#) 20

M

Message processing
[server](#) 13
Messages

[attribute groups](#) 12
[attributes](#) 12
[complex types](#) 12
[elements](#) 11
[enumerated](#) 11
[groups](#) 12
[namespaces](#) 11
[simple types](#) 12
[syntax](#) 11
[transport](#) 11

N

[Namespaces](#) 11
[Normative references](#) 7

O

Operations
[ResolveNames Operation](#) 13
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 22
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 27
Protocol Details
[overview](#) 13

R

[References](#) 7
[informative](#) 8
[normative](#) 7
[Relationship to other protocols](#) 8
[ResolveNames operation example](#) 21

S

Security
[implementer considerations](#) 22
[parameter index](#) 22
Sequencing rules
[server](#) 13
Server
[abstract data model](#) 13
[initialization](#) 13
[local events](#) 20
[message processing](#) 13
[ResolveNames Operation operation](#) 13
[sequencing rules](#) 13
[timer events](#) 20
[timers](#) 13
[Simple types](#) 12
[Standards assignments](#) 10
Syntax
[messages - overview](#) 11

T

Timer events

[server](#) 20

Timers

[server](#) 13

[Tracking changes](#) 28

[Transport](#) 11

Types

[complex](#) 12

[simple](#) 12

V

[Vendor-extensible fields](#) 10

[Versioning](#) 9

W

[WSDL](#) 23

X

[XML schema](#) 25

[Messages Schema](#) 25

[Types Schema](#) 26

Preliminary