[MS-OXWSCONT]:
Contacts Web Service Protocol Specification

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	2.0.0	Major	Updated and revised the technical content.
05/05/2010	2.0.1	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document specifies the Contacts Web Service protocol.

1.1 Glossary

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

SOAP body
SOAP fault
SOAP header
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", April 2008.

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

[MS-OXWSCORE] Microsoft Corporation, "Core Items Web Service Protocol Specification", July 2009.

[MS-OXWSDLIST] Microsoft Corporation, "Distribution List Creation and Usage Web Service Protocol Specification", July 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, 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., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, BCP 47, 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", W3C Note, May 2000, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

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

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

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

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

1.2.2 Informative References

None.

1.3 Overview

The Contacts Web Service protocol provides the messages needed to create, get, update, delete, move, and copy contact items on the server.

1.4 Relationship to Other Protocols

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

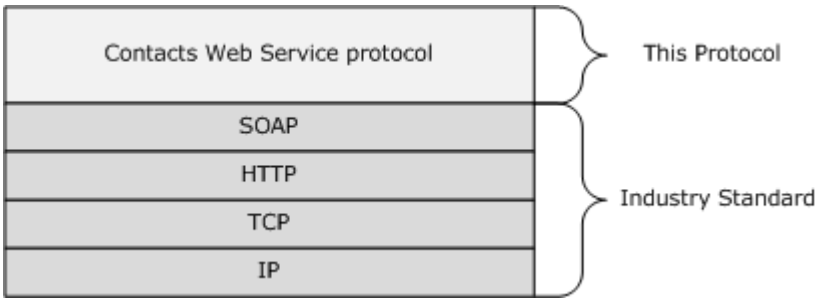

Figure 1: SOAP over HTTP

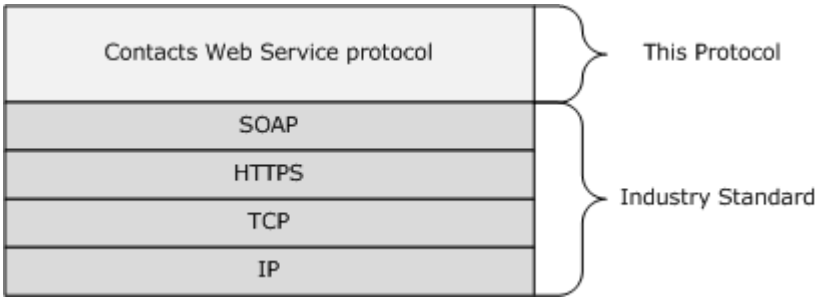

Figure 2: SOAP over HTTPS

1.5 Prerequisities/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable to client programs that create, update, or manage contact items in the server data store.

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, as specified in [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** 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-OXWSCONT]
s	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
targetNamespace	http://schemas.microsoft.com/exchange/services/2006/messages	[MS-OXWSCONT]
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	[MS-OXWSCONT]

2.2.2 Simple Types

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

Simple Type	Description
t:ContactSourceType	Specifies whether a contact or distribution list is located in the server database or in the directory service.

2.2.2.1 t:ContactSupportType Simple Type

The ContactSourceType specifies whether a contact or distribution list is located in the server database or in the directory service.

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

```
    base="xs:string"
  >
    <xs:enumeration
      value="ActiveDirectory"
     />
    <xs:enumeration
      value="Store"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

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

Value	Description
ActiveDirectory	Specifies that the contact or distribution list is located in the directory service.
Store	Specifies that the contact or distribution list is located in the server database.

This is applicable to [MS-OXWSCONT and [MS-OXWSDLIST] .

It is also used by the ResolveNames method, returning directory and store contacts matching a search string.

2.2.3 Complex Types

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

Complex Type	Description
t:ContactItemType	Represents a server contact item.

2.2.3.1 t:ContactItemType Complex Type

The **ContactItemType** complex type represents an Exchange contact item. It is also used by ResolveNames method, returning directory and store contacts matching a search string.

```
<xs:complexType name="t:ContactItemType">
  <xs:complexContent>
    <xs:extension
      base="t:ItemType"
    >
      <xs:sequence>
        <xs:element name="FileAs"
          type="xs:string"
          minOccurs="0"
         />
        <xs:element name="FileAsMapping"
          type="t:FileAsMappingType"
         />
```

```
<xs:element name="DisplayName"
  type="xs:string"
 />
<xs:element name="GivenName"
  type="xs:string"
 />
<xs:element name="Initials"
  type="xs:string"
 />
<xs:element name="MiddleName"
  type="xs:string"
 />
<xs:element name="Nickname"
  type="xs:string"
 />
<xs:element name="CompleteName"
  type="t:CompleteNameType"
 />
<xs:element name="CompanyName"
  type="xs:string"
 />
<xs:element name="EmailAddresses"
  type="t:EmailAddressDictionaryType"
 />
<xs:element name="PhysicalAddresses"
  type="t:PhysicalAddressDictionaryType"
 />
<xs:element name="PhoneNumbers"
  type="t:PhoneNumberDictionaryType"
 />
<xs:element name="AssistantName"
  type="xs:string"
 />
<xs:element name="Birthday"
  type="xs:dateTime"
 />
<xs:element name="BusinessHomePage"
  type="xs:anyURI"
 />
<xs:element name="Children"
  type="t:ArrayOfStringsType"
 />
<xs:element name="Companies"
  type="t:ArrayOfStringsType"
 />
<xs:element name="ContactSource"
  type="t:ContactSourceType"
 />
<xs:element name="Department"
  type="xs:string"
 />
<xs:element name="Generation"
  type="xs:string"
 />
<xs:element name="ImAddresses"
  type="t:ImAddressDictionaryType"
 />
<xs:element name="JobTitle"
  type="xs:string"
```

```
          />
        <xs:element name="Manager"
          type="xs:string"
         />
        <xs:element name="Mileage"
          type="xs:string"
         />
        <xs:element name="OfficeLocation"
          type="xs:string"
         />
        <xs:element name="PostalAddressIndex"
          type="t:PhysicalAddressIndexType"
         />
        <xs:element name="Profession"
          type="xs:string"
         />
        <xs:element name="SpouseName"
          type="xs:string"
         />
        <xs:element name="Surname"
          type="xs:string"
         />
        <xs:element name="WeddingAnniversary"
          type="xs:dateTime"
         />
        <xs:element name="HasPicture"
          type="xs:boolean"
         />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element	Type	Description
FileAs	xs:string	Represents how a contact is filed in the Contacts folder.
FileAsMapping	t:FileAsMappingType	Defines how to construct what is displayed for a contact.
DisplayName	xs:string	Contains the display name of a contact.
GivenName	xs:string	Contains the given name for a contact.
Initials	xs:string	Contains the initials for a contact.
MiddleName	xs:string	Represents the middle name of a contact.
Nickname	xs:string	Represents the nickname of a contact.
CompleteName	t:CompleteNameType	Represents the complete name of a contact.
CompanyName	xs:string	Contains the company name that is

Element	Type	Description
		associated with a contact.
EmailAddresses	t:EmailAddressDictionaryType	Contains e-mail addresses that are associated with a contact.
PhysicalAddresses	t:PhysicalAddressDictionaryType	Represents a collection of physical addresses that are associated with a contact.
PhoneNumbers	t:PhoneNumberDictionaryType	Represents a collection of telephone numbers for a contact.
AssistantName	xs:string	Contains the name of the assistant for the contact.
Birthday	xs:dateTime	Represents the birthday of the contact.
BusinessHomePage	xs:anyURI	Contains the business home page URL of a contact.
Children	t:ArrayOfStringsType	Contains the names of children for the contact.
Companies	**t:ArrayOfStringsType**	Contains the names of companies that are associated with a contact.
ContactSource	t:ContactSourceType	Describes whether the contact is located in the server store or the Active Directory service.
Department	xs:string	Contains the work department for the contact.
Generation	xs:string	Contains a generational abbreviation that follows the full name of a contact.
ImAddresses	t:ImAddressDictionaryType	Contains instant messaging addresses for a contact.
JobTitle	xs:string	Contains the job title of a contact.
Manager	xs:string	Represents the manager of a contact.
Mileage	xs:string	Represents the mileage for a contact.
OfficeLocation	xs:string	Represents the office location of a contact.
PostalAddressIndex	t:PhysicalAddressIndexType	Represents the index of one of the physical addresses, which is a contact's mailing address..
Profession	xs:string	Represents the profession of a contact.
SpouseName	xs:string	Represents the name of the spouse/partner of a contact.
Surname	xs:string	Contains the surname of a contact.

Element	Type	Description
WeddingAnniversary	xs:dateTime	Contains the wedding anniversary date of a contact.
HasPicture	xs:boolean	Represents that the contact has a picture.

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 Message Syntax

This specification does not define any common XML schema message syntax 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 returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 ExchangeServicePortType Server Details

This protocol defines a single port type.

Operation	Description
GetItem	Defines a request to get an item from a mailbox in the server.
DeleteItem	Defines a request to delete an item from a mailbox in the server.
UpdateItem	Defines a request to update an item in a mailbox.
MoveItem	Defines a request to move an item in the server.
CopyItem	Defines a request to copy an item in a mailbox in the server.
CreateItem	Defines a request to create an item in the server.

3.1.1 Server Abstract Data Model

This protocol is a stateless protocol.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Server Message Processing Events and Sequencing

This protocol includes the operations listed in the following table.

Operation	Description
GetItem	Gets items in the server.
DeleteItem	Deletes items in the server.
UpdateItem	Updates items in the server.
MoveItem	Moves items in the server.
CopyItem	Copies items in the server.
CreateItem	Creates items in the server.

3.1.4.1 GetItem

This protocol uses the **GetItem** operation specified in [MS-OXWSCORE] section 3.1.4.4 to get contact item elements.

```
<wsdl:operation name="GetItem">
          <wsdl:input message="tns:GetItemSoapIn" />
          <wsdl:output message="tns:GetItemSoapOut" />
      </wsdl:operation>
```

Request

Message Format	Description
tns:GetItemSoapIn	Specifies the SOAP message that defines the calendar item to get. The **Items** ([MS-OXWSCDATA] section 2.2.3.38) child element of the **GetItem** ([MS-OXWSCORE] section 3.1.4.4.2.1) child element that specifies the XML request MUST contain the following elements: **t:ItemResponseShapeType** ([MS-OXWSCDATA] section 2.2.3.35), **t:ItemIdType** ([MS-OXWSCORE] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:GetItemSoapOut	Specifies the [SOAP] message returned by the server in response. The server returns a **t:ItemResponseShapeType** ([MS-OXWSCDATA] section 2.2.3.35) element that contains properties associated with the distribution list item.

3.1.4.1.1 GetItem Simple Types

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

3.1.4.1.1.1 t:EmailAddressKeyType Simple Type

The **EmailAddressKeyType** simple type represents a way to identify a single e-mail address within the e-mail address collection for a contact.

```
<xs:simpleType name="EmailAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="EmailAddress1"
     />
    <xs:enumeration
      value="EmailAddress2"
     />
    <xs:enumeration
      value="EmailAddress3"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

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

Value	Description
EmailAddress1	Identifies the first e-mail address for the contact.
EmailAddress2	Identifies the second e-mail address for the contact.
EmailAddress3	Identifies the third e-mail address for the contact.

3.1.4.1.1.2 t:FileAsMappingType Simple Type

The **FileAsMappingType** simple type defines how to construct what is displayed for a contact in the **FileAs** property.

```
<xs:simpleType name="FileAsMappingType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="LastCommaFirst"
     />
    <xs:enumeration
      value="FirstSpaceLast"
     />
    <xs:enumeration
      value="Company"
     />
    <xs:enumeration
      value="LastCommaFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastFirst"
     />
    <xs:enumeration
      value="LastFirst"
     />
    <xs:enumeration
      value="LastFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastCommaFirst"
     />
    <xs:enumeration
      value="LastFirstSuffix"
     />
    <xs:enumeration
      value="LastSpaceFirstCompany"
     />
    <xs:enumeration
      value="CompanyLastSpaceFirst"
     />
```

```
      <xs:enumeration
        value="LastSpaceFirst"
       />
      <xs:enumeration
        value="DisplayName"
       />
      <xs:enumeration
        value="FirstName"
       />
      <xs:enumeration
        value="LastFirstMiddleSuffix"
       />
      <xs:enumeration
        value="LastName"
       />
      <xs:enumeration
        value="Empty"
       />
    </xs:restriction>
  </xs:simpleType>
```

Enumeration

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

Value	Description
None	Indicates that the **FileAs** value is not constructed from other contact's properties, but is represented by a string, saved "as is".
LastCommaFirst	Indicates that the contact is displayed as the last name followed by a comma and the first name.
FirstSpaceLast	Indicates that the contact is displayed as the first name followed by a space and the last name.
Company	Indicates that the company name is displayed.
LastCommaFirstCompany	Indicates that the contact is displayed as the last name, a comma, the first name, and the company name.
CompanyLastFirst	Indicates that the contact is displayed as the company name, the last name, and the first name.
LastFirst	Indicates that the contact is displayed as the last name followed by the first name.
LastFirstCompany	Indicates that the contact is displayed as the last name, the first name, and the company name.
CompanyLastCommaFirst	Indicates that the contact is displayed as the company name, the last name, a comma, and the first name.
LastFirstSuffix	Indicates that the contact is displayed as the last name, the first name, and the suffix for the contact.

Value	Description
LastSpaceFirstCompany	Indicates that the contact is displayed as the last name, a space, the first name, and the company name.
CompanyLastSpaceFirst	Indicates that the contact is displayed as the company name, the last name, a space, and the first name.
LastSpaceFirst	Indicates that the contact is displayed as the last name, followed by a space and the first name.
DisplayName	Indicates that the contact is displayed as the display name.
FirstName	Indicates that the contact is displayed as the first name.
LastFirstMiddleSuffix	Indicates that the contact is displayed as the last name, the first name, the middle name, and the suffix for the contact.
LastName	Indicates that the contact is displayed as the last name.
Empty	Indicates that the contact is displayed as empty.

3.1.4.1.1.3 t:ImAddressKeyType Simple Type

The **ImAddressKeyType** enumeration represents the instant messaging addresses for a contact.

```
<xs:simpleType name="ImAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="ImAddress1"
     />
    <xs:enumeration
      value="ImAddress2"
     />
    <xs:enumeration
      value="ImAddress3"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

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

Value	Description
ImAddress1	Identifies the first instant messaging address for the user.
ImAddress2	Identifies the second instant messaging address for the user.
ImAddress3	Identifies the third instant messaging address for the user.

3.1.4.1.1.4 t:PhoneNumberKeyType Simple Type

The **PhoneNumberKeyType** simple type represents types of telephone numbers for a contact.

```
<xs:simpleType name="PhoneNumberKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="AssistantPhone"
     />
    <xs:enumeration
      value="BusinessFax"
     />
    <xs:enumeration
      value="BusinessPhone"
     />
    <xs:enumeration
      value="BusinessPhone2"
     />
    <xs:enumeration
      value="Callback"
     />
    <xs:enumeration
      value="CarPhone"
     />
    <xs:enumeration
      value="CompanyMainPhone"
     />
    <xs:enumeration
      value="HomeFax"
     />
    <xs:enumeration
      value="HomePhone"
     />
    <xs:enumeration
      value="HomePhone2"
     />
    <xs:enumeration
      value="Isdn"
     />
    <xs:enumeration
      value="MobilePhone"
     />
    <xs:enumeration
      value="OtherFax"
     />
    <xs:enumeration
      value="OtherTelephone"
     />
    <xs:enumeration
      value="Pager"
     />
    <xs:enumeration
      value="PrimaryPhone"
     />
    <xs:enumeration
      value="RadioPhone"
     />
```

```
      <xs:enumeration
        value="Telex"
       />
      <xs:enumeration
        value="TtyTddPhone"
       />
    </xs:restriction>
  </xs:simpleType>
```

Enumeration

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

Value	Description
AssistantPhone	Identifies the telephone number as the assistant's telephone number.
BusinessFax	Identifies the telephone number as a business fax number.
BusinessPhone	Identifies the telephone number as a business telephone number.
BusinessPhone2	Identifies the telephone number as a second business telephone number.
Callback	Identifies the telephone number as a callback number.
CarPhone	Identifies the telephone number as a car telephone number.
CompanyMainPhone	Identifies the telephone number as the company's main telephone number.
HomeFax	Identifies the telephone number as a home fax number.
HomePhone	Identifies the telephone number as a home telephone number.
HomePhone2	Identifies the telephone number as a second home telephone number.
Isdn	Identifies the telephone number as an Integrated Services Digital Network (ISDN) line.
MobilePhone	Identifies the telephone number as a mobile phone number.
OtherFax	Identifies the telephone number as another fax number.
OtherTelephone	Identifies the telephone number as another telephone number.
Pager	Identifies the telephone number as a pager.
PrimaryPhone	Identifies the telephone number as the primary telephone number.
RadioPhone	Identifies the telephone number as a radio telephone.
Telex	Identifies the telephone number as a telex telephone number.
TtyTddPhone	Identifies the telephone number as a teletype/telecommunication device for the deaf (TTY/TDD) telephone number.

3.1.4.1.1.5 t:PhysicalAddressIndexType Simple Type

The **PhysicalAddressIndexType** complex type identifies the display types for physical addresses.

```
<xs:simpleType name="PhysicalAddressIndexType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="None"
     />
    <xs:enumeration
      value="Business"
     />
    <xs:enumeration
      value="Home"
     />
    <xs:enumeration
      value="Other"
     />
  </xs:restriction>
</xs:simpleType>
```

Enumeration

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

Value	Description
None	Indicates that no type is specified for the address.
Business	Displays the address as a business address.
Home	Displays the address as a home address.
Other	Displays the address as an address of type other.

3.1.4.1.1.6 t:PhysicalAddressKeyType Simple Type

The **PhysicalAddressKeyType** simple type identifies the types of physical addresses for a contact.

```
<xs:simpleType name="PhysicalAddressKeyType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="Business"
     />
    <xs:enumeration
      value="Home"
     />
    <xs:enumeration
      value="Other"
     />
  </xs:restriction>
```

```
</xs:simpleType>
```

Enumeration

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

Value	Description
Business	Identifies the address as a business address.
Home	Identifies the address as a home address.
Other	Identifies the address as an address of type other.

3.1.4.1.2 GetItem Complex Types

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

3.1.4.1.2.1 t:CompleteNameType Complex Type

The **CompleteNameType** complex type represents the complete name of a contact.

```
<xs:complexType name="CompleteNameType">
  <xs:sequence>
    <xs:element name="Title"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="FirstName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="MiddleName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="LastName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Suffix"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Initials"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="FullName"
      type="xs:string"
      minOccurs="0"
     />
    <xs:element name="Nickname"
      type="xs:string"
      minOccurs="0"
```

```
      />
     <xs:element name="YomiFirstName"
       type="xs:string"
       minOccurs="0"
      />
     <xs:element name="YomiLastName"
       type="xs:string"
       minOccurs="0"
      />
   </xs:sequence>
 </xs:complexType>
```

Child Elements

Element	Type	Description
Title	xs:string	Contains the title of a contact.
FirstName	xs:string	Contains the first name of a contact. This is the same as GivenName.
MiddleName	xs:string	Contains the middle name of a contact.
LastName	xs:string	Contains the last name of a contact. This is the same as the Surname.
Suffix	xs:string	Contains a suffix to a contact's name. This is the same as the Generation property.
Initials	xs:string	Contains the initials of a contact.
FullName	xs:string	Contains the full name of a contact.
Nickname	xs:string	Contains the nickname of a contact.
YomiFirstName	xs:string	Contains the name used in Japan for the searchable or phonetic spelling of a Japanese first name.
YomiLastName	xs:string	Contains the name used in Japan for the searchable or phonetic spelling of a Japanese last name.

3.1.4.1.2.2 t:ContactsFolderType Complex Type

The **ContactsFolderType** complex type represents a Contacts folder in a mailbox.

```
<xs:complexType name="ContactsFolderType">
  <xs:complexContent>
    <xs:extension
      base="t:BaseFolderType"
    >
      <xs:sequence>
        <xs:element name="SharingEffectiveRights"
          type="t:PermissionReadAccessType"
          minOccurs="0"
         />
        <xs:element name="PermissionSet"
          type="t:PermissionSetType"
          minOccurs="0"
```

```
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Child Elements

Element	Type	Description
SharingEffectiveRights	t:PermissionReadAccessType	Specifies whether a user has permission to read items in a folder.
PermissionSet	t:PermissionSetType	Contains all the permissions that are configured for a folder.

3.1.4.1.2.3 t:ContactsViewType Complex Type

The **ContactsViewType** complex type represents the settings that are used to return contact items based on their alphabetical display names.

```
<xs:complexType name="ContactsViewType">
  <xs:complexContent>
    <xs:extension
      base="t:BasePagingType"
    >
      <xs:attribute name="InitialName"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="FinalName"
        type="xs:string"
        use="optional"
       />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Attributes

Name	Type	Description
InitialName	xs:string	Contains the first name in a contacts list to return in a response.
FinalName	xs:string	Contains the last name in a contacts list to return in a response.

3.1.4.1.2.4 t:EmailAddressDictionaryEntryType Complex Type

The **EmailAddressDictionaryEntryType** complex type represents an e-mail address that is associated with a contact.

```
<xs:complexType>
```

```
  <xs:simpleContent>
    <xs:extension
      base="xs:string"
    >
      <xs:attribute name="Key"
        type="t:EmailAddressKeyType"
        use="required"
       />
      <xs:attribute name="Name"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="RoutingType"
        type="xs:string"
        use="optional"
       />
      <xs:attribute name="MailboxType"
        type="t:MailboxTypeType"
        use="optional"
       />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Attributes

Name	Type	Description
Key	t:EmailAddressKeyType	Contains a value that identifies an e-mail address that is associated with a contact.
Name	xs:string	Contains the display name associated with an e-mail address of the contact.
RoutingType	xs:string	Contains the routing type associated with an e-mail address of the contact.
MailboxType	t:MailboxTypeType	Contains the type of mailbox that is represented by the e-mail address of the contact.

3.1.4.1.2.5 t:EmailAddressDictionaryType Complex Type

The EmailAddressDictionaryType complex type contains e-mail addresses.

```
<xs:complexType name="EmailAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:EmailAddressDictionaryEntryType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element	Type	Description
Entry	t:EmailAddressDictionaryEntryType	Represents an e-mail address that is associated with a contact.

3.1.4.1.2.6 t:ImAddressDictionaryEntryType Complex Type

The **ImAddressDictionaryEntryType** complex type represents a collection of instant messaging addresses for a contact.

```
<xs:complexType name="ImAddressDictionaryEntryType">
  <xs:simpleContent>
    <xs:extension
      base="xs:string"
    >
      <xs:attribute name="key"
        type="t:ImAddressKeyType"
        use="required"
       />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Attributes

Name	Type	Description
key	t:ImAddressKeyType	Represents the instant messaging addresses for a contact.

3.1.4.1.2.7 t:ImAddressDictionaryType Complex Type

The **ImAddressDictionaryType** complex type contains instant messaging addresses for a contact.

```
<xs:complexType name="ImAddressDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:ImAddressDictionaryEntryType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element	Type	Description
Entry	t:ImAddressDictionaryEntryType	Represents a collection of instant messaging addresses for a contact.

3.1.4.1.2.8 t:PhoneNumberDictionaryEntryType Complex Type

The **PhoneNumberDictionaryEntryType** complex type contains a telephone number for a contact.

```
<xs:complexType name="PhoneNumberDictionaryEntryType">
  <xs:simpleContent>
    <xs:extension
      base="xs:string"
    >
      <xs:attribute name="Key"
        type="t:PhoneNumberKeyType"
        use="required"
       />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Attributes

Name	Type	Description
Key	t:PhoneNumberKeyType	Represents types of telephone numbers for a contact.

3.1.4.1.2.9 t:PhoneNumberDictionaryType Complex Type

The **PhoneNumberDictionaryType** complex type represents telephone numbers for a contact.

```
<xs:complexType name="PhoneNumberDictionaryType">
  <xs:sequence>
    <xs:element name="Entry"
      type="t:PhoneNumberDictionaryEntryType"
      maxOccurs="unbounded"
     />
  </xs:sequence>
</xs:complexType>
```

Child Elements

Element	Type	Description
Entry	t:PhoneNumberDictionaryEntryType	Contains a telephone number for a contact.

3.1.4.1.2.10 t:PhysicalAddressDictionaryEntryType Complex Type

The **PhysicalAddressDictionaryEntryType** complex type contains information that defines a physical address, such as a street address.

```
<xs:complexType name="PhysicalAddressDictionaryEntryType">
  <xs:sequence>
    <xs:element name="Street"
      type="xs:string"
      minOccurs="0"
```

```
      />
     <xs:element name="City"
       type="xs:string"
       minOccurs="0"
      />
     <xs:element name="State"
       type="xs:string"
       minOccurs="0"
      />
     <xs:element name="CountryOrRegion"
       type="xs:string"
       minOccurs="0"
      />
     <xs:element name="PostalCode"
       type="xs:string"
       minOccurs="0"
      />
   </xs:sequence>
   <xs:attribute name="Key"
     type="t:PhysicalAddressKeyType"
     use="required"
    />
</xs:complexType>
```

Child Elements

Element	Type	Description
Street	xs:string	Contains the street address for a contact item.
City	xs:string	Contains the city name for a contact item.
State	xs:string	Contains the state for a contact item.
CountryOrRegion	xs:string	Contains the country or region for a contact item.
PostalCode	xs:string	Contains the postal code for a contact item.

Attributes

Name	Type	Description
Key	t:PhysicalAddressKeyType	Identifies the types of physical addresses for a contact.

3.1.4.1.2.11 t:PhysicalAddressDictionaryType Complex Type

The **PhysicalAddressDictionaryType** complex type contains physical addresses that are associated with a contact.

```
<xs:complexType name="PhysicalAddressDictionaryType">
  <xs:sequence>
    <xs:element name="entry"
      type="t:PhysicalAddressDictionaryEntryType"
      maxOccurs="unbounded"
     />
```

```
    </xs:sequence>
  </xs:complexType>
```

Child Elements

Element	Type	Description
entry	**t:PhysicalAddressDictionaryEntryType**	Contains information that defines a physical address, such as a street address.

3.1.4.2 DeleteItem

This protocol uses the **DeleteItem** operation specified in [MS-OXWSCORE] section 3.1.4.3 to delete contact item elements.

```
<wsdl:operation name="DeleteItem">
            <wsdl:input message="tns:DeleteItemSoapIn" />
            <wsdl:output message="tns:DeleteItemSoapOut" />
            </wsdl:operation>
```

Request

Message Format	Description
tns: DeleteItemSoapIn	Specifies the SOAP message that defines the contact item to delete. The Items child element of the DeleteItem child element that specifies the XML request MUST contain one or more t:ItemIdType ([MS-OXWSCORE] section 2.2.3.13) elements. All other elements MUST be empty.

Response

Message Format	Description
tns:DeleteItemSoapOut	Specifies the [SOAP] message returned by the server in response.

3.1.4.3 UpdateItem

This protocol uses the **UpdateItem** operation specified in [MS-OXWSCORE] section 3.1.4.7 to update contact item elements.

```
<wsdl:operation name="UpdateItem">
            <wsdl:input message="tns:UpdateItemSoapIn" />
            <wsdl:output message="tns:UpdateItemSoapOut" />
        </wsdl:operation>
```

Request

Message Format	Description
tns:UpdateItemSoapIn	Specifies the SOAP message that defines the contact item to update. The

Message Format	Description
	Items child element of the UpdateItem child element that specifies the XML request MUST contain one or more t:ContactItemType elements. All other elements MUST be empty.

Response

Message Format	Description
tns:UpdateItemSoapOut	Specifies the [SOAP] message returned by the server in response.

3.1.4.4 MoveItem

This protocol uses the **MoveItem** operation specified in [MS-OXWSCORE] section 3.1.4.5 to move contact item elements.

```
<wsdl:operation name="MoveItem">
            <wsdl:input message="tns:MoveItemSoapIn" />
            <wsdl:output message="tns:MoveItemSoapOut" />
            </wsdl:operation>
```

Request

Message Format	Description
tns:MoveItemSoapIn	Specifies the SOAP message that defines the contact item to move. The Items child element of the MoveItem child element that specifies the XML request MUST contain the following elements: t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.3.14), and t:ItemIdType ([MS-OXWSCORE] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:MoveItemSoapOut	Specifies the [SOAP] message returned by the server in response.

3.1.4.5 CopyItem

This protocol uses the **CopyItem** operation specified in [MS-OXWSCORE] section 3.1.4.1 to copy contact item elements.

```
<wsdl:operation name="CopyItem">
            <wsdl:input message="tns:CopyItemSoapIn" />
            <wsdl:output message="tns:CopyItemSoapOut" />
            </wsdl:operation>
```

Request

Message Format	Description
tns:CopyItemSoapIn	Specifies the SOAP message that defines the contact item to copy. The Items child element of the CopyItem child element that specifies the XML request MUST contain the following elements: t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.3.14), and t:ItemIdType ([MS-OXWSCORE] section 2.2.3.13). All other elements MUST be empty.

Response

Message Format	Description
tns:CopyItemSoapOut	Specifies the [SOAP] message returned by the server in response.

3.1.4.6 CreateItem

This protocol uses the **CreateItem** operation specified in [MS-OXWSCORE] section 3.1.4.2 to create contact item elements.

```
<wsdl:operation name="CreateItem">
     <wsdl:input message="tns:CreateItemSoapIn" />
     <wsdl:output message="tns:CreateItemSoapOut" />
</wsdl:operation>
```

Request

Message Format	Description
tns:CreateItemSoapIn	Specifies the SOAP message that defines the contact item to create. The Items child element of the CreateItem child element that specifies the XML request MUST contain one or more t:ContactItemType elements. All other elements MUST be empty.

Response

Message Format	Description
tns:CreateItemSoapOut	Specifies the [SOAP] message returned by the server in response.

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

3.2.1 Client Abstract Data Model

None.

3.2.2 Client Timers

None.

3.2.3 Client Initialization

None.

3.2.4 Client Message Processing Events and Sequencing

None.

3.2.5 Client Timer Events

None.

3.2.6 Client Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementors

This protocol does not use additional security mechanisms.

5.2 Index of Security Parameters

None.

6 Appendix A: Full WSDL

The following table lists the **XML** files that are required 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-OXWSCONT.wsdl	Contains the WSDL for the implementation of this protocol.	6.1
MS-OXWSCONT-types.xsd	Contains the XML schema type definitions that are used in this protocol.	6.2
MS-OXWSCONT-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	6.3

These files need to be placed in a common folder for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSGTRM-types.xsd or MS-OXWSGTRM-messages.xsd schemas need to be placed in the common folder with these files.

6.1 WSDL

This section contains the content of the MS-OXWSCONT.wsdl file and information about additional files that this WSDL file requires to operate correctly.

For MS-OXWSCONT.wsdl to operate correctly, the file shown in the following table needs to be present in the folder that contains the WSDL, types schema and messages schema files for this protocol.

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

MS-OXWSCONT.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:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
               <xs:include schemaLocation="MS-OXWSCORE-messages.xsd"/>
          </xs:schema>
     </wsdl:types>
     <wsdl:portType name="ExchangeServicePortType">
          <wsdl:operation name="GetItem">
               <wsdl:input message="tns:GetItemSoapIn"/>
               <wsdl:output message="tns:GetItemSoapOut"/>
          </wsdl:operation>
          <wsdl:operation name="CreateItem">
               <wsdl:input message="tns:CreateItemSoapIn"/>
               <wsdl:output message="tns:CreateItemSoapOut"/>
          </wsdl:operation>
```

```
            <wsdl:operation name="DeleteItem">
                 <wsdl:input message="tns:DeleteItemSoapIn"/>
                 <wsdl:output message="tns:DeleteItemSoapOut"/>
            </wsdl:operation>
            <wsdl:operation name="UpdateItem">
                 <wsdl:input message="tns:UpdateItemSoapIn"/>
                 <wsdl:output message="tns:UpdateItemSoapOut"/>
            </wsdl:operation>
            <wsdl:operation name="MoveItem">
                 <wsdl:input message="tns:MoveItemSoapIn"/>
                 <wsdl:output message="tns:MoveItemSoapOut"/>
            </wsdl:operation>
            <wsdl:operation name="CopyItem">
                 <wsdl:input message="tns:CopyItemSoapIn"/>
                 <wsdl:output message="tns:CopyItemSoapOut"/>
            </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="GetItem">
                 <soap:operation
  soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetItem"/>
                 <wsdl:input>
                      <soap:header message="tns:GetItemSoapIn" part="Impersonation"
  use="literal"/>
                      <soap:header message="tns:GetItemSoapIn" part="MailboxCulture"
  use="literal"/>
                      <soap:header message="tns:GetItemSoapIn" part="RequestVersion"
  use="literal"/>
                      <soap:header message="tns:GetItemSoapIn" part="TimeZoneContext"
  use="literal"/>
                      <soap:body parts="request" use="literal"/>
                 </wsdl:input>
                 <wsdl:output>
                      <soap:body parts="GetItemResult" use="literal"/>
                      <soap:header message="tns:GetItemSoapOut" part="ServerVersion"
  use="literal"/>
                 </wsdl:output>
            </wsdl:operation>
            <wsdl:operation name="CreateItem">
                 <soap:operation
  soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CreateItem"/>
                 <wsdl:input>
                      <soap:header message="tns:CreateItemSoapIn" part="Impersonation"
  use="literal"/>
                      <soap:header message="tns:CreateItemSoapIn" part="MailboxCulture"
  use="literal"/>
                      <soap:header message="tns:CreateItemSoapIn" part="RequestVersion"
  use="literal"/>
                      <soap:header message="tns:CreateItemSoapIn" part="TimeZoneContext"
  use="literal"/>
                      <soap:body parts="request" use="literal"/>
                 </wsdl:input>
                 <wsdl:output>
                      <soap:body parts="CreateItemResult" use="literal"/>
```

```
                    <soap:header message="tns:CreateItemSoapOut" part="ServerVersion"
use="literal"/>
               </wsdl:output>
          </wsdl:operation>
          <wsdl:operation name="DeleteItem">
               <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/DeleteItem"/>
               <wsdl:input>
                    <soap:header message="tns:DeleteItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:DeleteItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:DeleteItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
               </wsdl:input>
               <wsdl:output>
                    <soap:body parts="DeleteItemResult" use="literal"/>
                    <soap:header message="tns:DeleteItemSoapOut" part="ServerVersion"
use="literal"/>
               </wsdl:output>
          </wsdl:operation>
          <wsdl:operation name="UpdateItem">
               <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/UpdateItem"/>
               <wsdl:input>
                    <soap:header message="tns:UpdateItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:UpdateItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:UpdateItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:header message="tns:UpdateItemSoapIn" part="TimeZoneContext"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
               </wsdl:input>
               <wsdl:output>
                    <soap:body parts="UpdateItemResult" use="literal"/>
                    <soap:header message="tns:UpdateItemSoapOut" part="ServerVersion"
use="literal"/>
               </wsdl:output>
          </wsdl:operation>
          <wsdl:operation name="MoveItem">
               <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/MoveItem"/>
               <wsdl:input>
                    <soap:header message="tns:MoveItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:MoveItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:MoveItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
               </wsdl:input>
               <wsdl:output>
                    <soap:body parts="MoveItemResult" use="literal"/>
                    <soap:header message="tns:MoveItemSoapOut" part="ServerVersion"
use="literal"/>
               </wsdl:output>
          </wsdl:operation>
```

```
          <wsdl:operation name="CopyItem">
               <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/CopyItem"/>
               <wsdl:input>
                    <soap:header message="tns:CopyItemSoapIn" part="Impersonation"
use="literal"/>
                    <soap:header message="tns:CopyItemSoapIn" part="MailboxCulture"
use="literal"/>
                    <soap:header message="tns:CopyItemSoapIn" part="RequestVersion"
use="literal"/>
                    <soap:body parts="request" use="literal"/>
               </wsdl:input>
               <wsdl:output>
                    <soap:body parts="CopyItemResult" use="literal"/>
                    <soap:header message="tns:CopyItemSoapOut" part="ServerVersion"
use="literal"/>
               </wsdl:output>
          </wsdl:operation>
     </wsdl:binding>
     <wsdl:message name="GetItemSoapIn">
          <wsdl:part name="request" element="tns:GetItem"/>
          <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
          <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
          <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
          <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
     </wsdl:message>
     <wsdl:message name="GetItemSoapOut">
          <wsdl:part name="GetItemResult" element="tns:GetItemResponse"/>
          <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
     </wsdl:message>
     <wsdl:message name="CreateItemSoapIn">
          <wsdl:part name="request" element="tns:CreateItem"/>
          <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
          <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
          <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
          <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
     </wsdl:message>
     <wsdl:message name="CreateItemSoapOut">
          <wsdl:part name="CreateItemResult" element="tns:CreateItemResponse"/>
          <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
     </wsdl:message>
     <wsdl:message name="DeleteItemSoapIn">
          <wsdl:part name="request" element="tns:DeleteItem"/>
          <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="DeleteItemSoapOut">
          <wsdl:part name="DeleteItemResult" element="tns:DeleteItemResponse"/>
          <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
     </wsdl:message>
     <wsdl:message name="UpdateItemSoapIn">
          <wsdl:part name="request" element="tns:UpdateItem"/>
          <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
          <wsdl:part name="MailboxCulture" element="t:MailboxCulture"/>
          <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
          <wsdl:part name="TimeZoneContext" element="t:TimeZoneContext"/>
     </wsdl:message>
     <wsdl:message name="UpdateItemSoapOut">
```

```
            <wsdl:part name="UpdateItemResult" element="tns:UpdateItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
      </wsdl:message>
      <wsdl:message name="MoveItemSoapIn">
            <wsdl:part name="request" element="tns:MoveItem"/>
            <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="MoveItemSoapOut">
            <wsdl:part name="MoveItemResult" element="tns:MoveItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
      </wsdl:message>
      <wsdl:message name="CopyItemSoapIn">
            <wsdl:part name="request" element="tns:CopyItem"/>
            <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="CopyItemSoapOut">
            <wsdl:part name="CopyItemResult" element="tns:CopyItemResponse"/>
            <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
      </wsdl:message>
</wsdl:definitions>
```

6.2 Types Schema

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

For MS-OXWSCONT-types.xsd to operate correctly, the file shown in the following table needs 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-types.xsd

MS-OXWSCONT-types.xsd file:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
elementFormDefault="qualified" version="Exchange2010" id="types">
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
      <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
      <xs:complexType name="CompleteNameType">
            <xs:sequence>
                  <xs:element name="Title" type="xs:string" minOccurs="0"/>
                  <xs:element name="FirstName" type="xs:string" minOccurs="0"/>
                  <xs:element name="MiddleName" type="xs:string" minOccurs="0"/>
                  <xs:element name="LastName" type="xs:string" minOccurs="0"/>
                  <xs:element name="Suffix" type="xs:string" minOccurs="0"/>
```

```
                <xs:element name="Initials" type="xs:string" minOccurs="0"/>
                <xs:element name="FullName" type="xs:string" minOccurs="0"/>
                <xs:element name="Nickname" type="xs:string" minOccurs="0"/>
                <xs:element name="YomiFirstName" type="xs:string" minOccurs="0"/>
                <xs:element name="YomiLastName" type="xs:string" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="ContactItemType">
            <xs:complexContent>
                <xs:extension base="t:ItemType">
                    <xs:sequence>
                        <xs:element name="FileAs" type="xs:string" minOccurs="0"/>
                        <xs:element name="FileAsMapping" type="t:FileAsMappingType"
minOccurs="0"/>
                        <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
                        <xs:element name="GivenName" type="xs:string" minOccurs="0"/>
                        <xs:element name="Initials" type="xs:string" minOccurs="0"/>
                        <xs:element name="MiddleName" type="xs:string" minOccurs="0"/>
                        <xs:element name="Nickname" type="xs:string" minOccurs="0"/>
                        <xs:element name="CompleteName" type="t:CompleteNameType"
minOccurs="0"/>
                        <xs:element name="CompanyName" type="xs:string" minOccurs="0"/>
                        <xs:element name="EmailAddresses"
type="t:EmailAddressDictionaryType" minOccurs="0"/>
                        <xs:element name="PhysicalAddresses"
type="t:PhysicalAddressDictionaryType" minOccurs="0"/>
                        <xs:element name="PhoneNumbers" type="t:PhoneNumberDictionaryType"
minOccurs="0"/>
                        <xs:element name="AssistantName" type="xs:string" minOccurs="0"/>
                        <xs:element name="Birthday" type="xs:dateTime" minOccurs="0"/>
                        <xs:element name="BusinessHomePage" type="xs:anyURI" minOccurs="0"/>
                        <xs:element name="Children" type="t:ArrayOfStringsType"
minOccurs="0"/>
                        <xs:element name="Companies" type="t:ArrayOfStringsType"
minOccurs="0"/>
                        <xs:element name="ContactSource" type="t:ContactSourceType"
minOccurs="0"/>
                        <xs:element name="Department" type="xs:string" minOccurs="0"/>
                        <xs:element name="Generation" type="xs:string" minOccurs="0"/>
                        <xs:element name="ImAddresses" type="t:ImAddressDictionaryType"
minOccurs="0"/>
                        <xs:element name="JobTitle" type="xs:string" minOccurs="0"/>
                        <xs:element name="Manager" type="xs:string" minOccurs="0"/>
                        <xs:element name="Mileage" type="xs:string" minOccurs="0"/>
                        <xs:element name="OfficeLocation" type="xs:string" minOccurs="0"/>
                        <xs:element name="PostalAddressIndex"
type="t:PhysicalAddressIndexType" minOccurs="0"/>
                        <xs:element name="Profession" type="xs:string" minOccurs="0"/>
                        <xs:element name="SpouseName" type="xs:string" minOccurs="0"/>
                        <xs:element name="Surname" type="xs:string" minOccurs="0"/>
                        <xs:element name="WeddingAnniversary" type="xs:dateTime"
minOccurs="0"/>
                        <xs:element name="HasPicture" type="xs:boolean" minOccurs="0"/>
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:complexType name="ContactsFolderType">
            <xs:complexContent>
                <xs:extension base="t:BaseFolderType">
```

```
                    <xs:sequence>
                         <xs:element name="SharingEffectiveRights"
type="t:PermissionReadAccessType" minOccurs="0"/>
                         <xs:element name="PermissionSet" type="t:PermissionSetType"
minOccurs="0"/>
                    </xs:sequence>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:simpleType name="ContactSourceType">
          <xs:restriction base="xs:string">
               <xs:enumeration value="ActiveDirectory"/>
               <xs:enumeration value="Store"/>
          </xs:restriction>
     </xs:simpleType>
     <xs:complexType name="ContactsViewType">
          <xs:complexContent>
               <xs:extension base="t:BasePagingType">
                    <xs:attribute name="InitialName" type="xs:string" use="optional"/>
                    <xs:attribute name="FinalName" type="xs:string" use="optional"/>
               </xs:extension>
          </xs:complexContent>
     </xs:complexType>
     <xs:complexType name="EmailAddressDictionaryEntryType">
          <xs:simpleContent>
               <xs:extension base="xs:string">
                    <xs:attribute name="Key" type="t:EmailAddressKeyType" use="required"/>
                    <xs:attribute name="Name" type="xs:string" use="optional"/>
                    <xs:attribute name="RoutingType" type="xs:string" use="optional"/>
                    <xs:attribute name="MailboxType" type="t:MailboxTypeType"
use="optional"/>
               </xs:extension>
          </xs:simpleContent>
     </xs:complexType>
     <xs:complexType name="EmailAddressDictionaryType">
          <xs:sequence>
               <xs:element name="Entry" type="t:EmailAddressDictionaryEntryType"
maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <xs:simpleType name="EmailAddressKeyType">
          <xs:restriction base="xs:string">
               <xs:enumeration value="EmailAddress1"/>
               <xs:enumeration value="EmailAddress2"/>
               <xs:enumeration value="EmailAddress3"/>
          </xs:restriction>
     </xs:simpleType>
     <xs:simpleType name="FileAsMappingType">
          <xs:restriction base="xs:string">
               <xs:enumeration value="None"/>
               <xs:enumeration value="LastCommaFirst"/>
               <xs:enumeration value="FirstSpaceLast"/>
               <xs:enumeration value="Company"/>
               <xs:enumeration value="LastCommaFirstCompany"/>
               <xs:enumeration value="CompanyLastFirst"/>
               <xs:enumeration value="LastFirst"/>
               <xs:enumeration value="LastFirstCompany"/>
               <xs:enumeration value="CompanyLastCommaFirst"/>
               <xs:enumeration value="LastFirstSuffix"/>
```

```
                <xs:enumeration value="LastSpaceFirstCompany"/>
                <xs:enumeration value="CompanyLastSpaceFirst"/>
                <xs:enumeration value="LastSpaceFirst"/>
                <xs:enumeration value="DisplayName"/>
                <xs:enumeration value="FirstName"/>
                <xs:enumeration value="LastFirstMiddleSuffix"/>
                <xs:enumeration value="LastName"/>
                <xs:enumeration value="Empty"/>

           </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="ImAddressDictionaryEntryType">
           <xs:simpleContent>
                <xs:extension base="xs:string">
                     <xs:attribute name="Key" type="t:ImAddressKeyType" use="required"/>
                </xs:extension>
           </xs:simpleContent>
      </xs:complexType>
      <xs:complexType name="ImAddressDictionaryType">
           <xs:sequence>
                <xs:element name="Entry" type="t:ImAddressDictionaryEntryType"
  maxOccurs="unbounded"/>
           </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="ImAddressKeyType">
           <xs:restriction base="xs:string">
                <xs:enumeration value="ImAddress1"/>
                <xs:enumeration value="ImAddress2"/>
                <xs:enumeration value="ImAddress3"/>
           </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="PhoneNumberDictionaryEntryType">
           <xs:simpleContent>
                <xs:extension base="xs:string">
                     <xs:attribute name="Key" type="t:PhoneNumberKeyType" use="required"/>
                </xs:extension>
           </xs:simpleContent>
      </xs:complexType>
      <xs:complexType name="PhoneNumberDictionaryType">
           <xs:sequence>
                <xs:element name="Entry" type="t:PhoneNumberDictionaryEntryType"
  maxOccurs="unbounded"/>
           </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="PhoneNumberKeyType">
           <xs:restriction base="xs:string">
                <xs:enumeration value="AssistantPhone"/>
                <xs:enumeration value="BusinessFax"/>
                <xs:enumeration value="BusinessPhone"/>
                <xs:enumeration value="BusinessPhone2"/>
                <xs:enumeration value="Callback"/>
                <xs:enumeration value="CarPhone"/>
                <xs:enumeration value="CompanyMainPhone"/>
                <xs:enumeration value="HomeFax"/>
                <xs:enumeration value="HomePhone"/>
                <xs:enumeration value="HomePhone2"/>
                <xs:enumeration value="Isdn"/>
                <xs:enumeration value="MobilePhone"/>
                <xs:enumeration value="OtherFax"/>
```

```
                <xs:enumeration value="OtherTelephone"/>
                <xs:enumeration value="Pager"/>
                <xs:enumeration value="PrimaryPhone"/>
                <xs:enumeration value="RadioPhone"/>
                <xs:enumeration value="Telex"/>
                <xs:enumeration value="TtyTddPhone"/>
           </xs:restriction>
      </xs:simpleType>
      <xs:complexType name="PhysicalAddressDictionaryEntryType">
           <xs:sequence>
                <xs:element name="Street" type="xs:string" minOccurs="0"/>
                <xs:element name="City" type="xs:string" minOccurs="0"/>
                <xs:element name="State" type="xs:string" minOccurs="0"/>
                <xs:element name="CountryOrRegion" type="xs:string" minOccurs="0"/>
                <xs:element name="PostalCode" type="xs:string" minOccurs="0"/>
           </xs:sequence>
           <xs:attribute name="Key" type="t:PhysicalAddressKeyType" use="required"/>
      </xs:complexType>
      <xs:complexType name="PhysicalAddressDictionaryType">
           <xs:sequence>
                <xs:element name="Entry" type="t:PhysicalAddressDictionaryEntryType"
  maxOccurs="unbounded"/>
           </xs:sequence>
      </xs:complexType>
      <xs:simpleType name="PhysicalAddressIndexType">
           <xs:restriction base="xs:string">
                <xs:enumeration value="None"/>
                <xs:enumeration value="Business"/>
                <xs:enumeration value="Home"/>
                <xs:enumeration value="Other"/>
           </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="PhysicalAddressKeyType">
           <xs:restriction base="xs:string">
                <xs:enumeration value="Business"/>
                <xs:enumeration value="Home"/>
                <xs:enumeration value="Other"/>
           </xs:restriction>
      </xs:simpleType>
  </xs:schema>
```

6.3 Messages Schema

No messages schema file is specified for this protocol.

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-OXWSCONT] protocol documentation between February 2010 and May 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
1.3 Overview	Updated the section title.	N	Content updated for template compliance.

9 Index