

[MS-OXOCNTC]: Contact Object Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol 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 protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- **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 protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp/default.msp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@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.

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.

Preliminary Documentation. This documentation is preliminary documentation for these protocols. Since the documentation may change between this preliminary version and the final version, 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.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and networking programming art, and assumes that the reader is either familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for a Licensee to develop an implementation. Licensees who have access to Microsoft programming tools and environments are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 4, 2008	0.1	Initial Availability

Preliminary

Table of Contents

1	Introduction	4
1.1	Glossary	4
1.2	References	5
1.2.1	Normative References.....	5
1.2.2	Informative References	6
1.3	Protocol Overview (Synopsis).....	6
1.4	Relationship to Other Protocols	7
1.5	Prerequisites/Preconditions	7
1.6	Applicability Statement.....	7
1.7	Versioning and Capability Negotiation	7
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments.....	7
2	Messages	7
2.1	Transport.....	7
2.2	Message Syntax.....	7
2.2.1	Contact Object Properties	8
2.2.2	Personal Distribution List Properties.....	31
2.2.3	Contact Folder Syntax.....	35
3	Protocol Details	35
3.1	Client and Server Details	35
3.1.1	Abstract Data Model	35
3.1.2	Timers	36
3.1.3	Initialization.....	36
3.1.4	Higher-Layer Triggered Events.....	37
3.1.5	Message Processing Events and Sequencing Rules	45
3.1.6	Timer Events	46
3.1.7	Other Local Events.....	46
4	Protocol Examples	46
4.1	Creating a Contact	46
4.2	Creating a Personal Distribution List	50
5	Security	53
5.1	Security Considerations for Implementers	53
5.2	Index of Security Parameters	53
6	Appendix A: Office/Exchange Behavior	53
7	Index	57

1 Introduction

This document specifies the Contact Object Protocol, which defines properties of objects that contain contact information, such as phone numbers, e-mail addresses, and mailing addresses. It also defines properties of objects containing collections of e-mail addresses, as might be used to define a mailing list.

This protocol extends the Message and Attachment Object Protocol Specification, as specified in [MS-OXCMSG], and specifies the properties that are set on **Contact objects** and on a **Personal Distribution List objects** in addition to the **Message object** properties as specified in [MS-OXCMSG]. This specification assumes the reader has familiarity with the concepts and requirements of the Message and Attachment Object Protocol Specification. Those concepts and requirements are not repeated in this specification.

1.1 Glossary

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

Address Book object

binary large object (BLOB)

Coordinated Universal Time (UTC)

folder

JPG

little-endian

locale

mailbox

Message object

one-off EntryId

PNG

portable network graphics (PNG)

property

store

Unicode

The following data types are defined in [MS-OXCDATA]:

PtypBinary

PtypBoolean

PtypFloatingTime

PtypInteger16

PtypInteger32

PtypMultipleBinary

PtypMultipleInteger32

PtypString

The following terms are specific to this document:

address type: An identifier for the type of e-mail address, such as SMTP and EX.

Contact object: A **Message object** that contains properties pertaining to a contact.

Personal Distribution List object: A **Message object** that contains properties that pertain specifically to a user-created distribution list.

point: A unit of length used in typography to specify text character height and width.

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

[ISO/IEC 8802-3] International Organization for Standardization, "Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks -- Specific requirements -- Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", ISO/IEC 8802-3:2000,

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31002

[MS-OXCDATA] Microsoft Corporation, "Data Structures Protocol Specification", April 2008.

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol Specification", April 2008.

[MS-OXCICAL] Microsoft Corporation, "iCalendar to Appointment Object Conversion Protocol Specification", April 2008.

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", April 2008.

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification", April 2008.

[MS-OXGLOS] Microsoft Corporation, "Office Exchange Protocols Master Glossary", April 2008.

[MS-OXOABK] Microsoft Corporation, "Address Book Object Protocol Specification", April 2008.

[MS-OXOCAL] Microsoft Corporation, "Appointment and Meeting Object Protocol Specification", April 2008.

[MS-OXPROPS] Microsoft Corporation, "Office Exchange Protocols Master Property List Specification", April 2008.

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

[RFC2247] Kille, S., Wahl, M., Grimstad, A., Huber, R., and Sataluri, S., "Using Domains in LDAP/X.500 Distinguished Names", RFC 2247, January 1998, <http://www.ietf.org/rfc/rfc2247.txt>

[RFC821] Postel, J., "Simple Mail Transfer Protocol", RFC 821, August 1982, <http://www.ietf.org/rfc/rfc821.txt>.

[RFC959] Postel, J., Reynolds, J., "File Transfer Protocol (FTP)", RFC 959, October 1985, <http://www.ietf.org/rfc/rfc959.txt>.

1.2.2 Informative References

None.

1.3 Protocol Overview (Synopsis)

A Contact object is a **Message object** that represents a person or an entity. A Personal Distribution List object is a Message object representing a group of e-mail addresses.

This protocol is an extension of the Message and Attachment Object Protocol Specification, as specified in [MS-OXCMSG], and specifies the representation of a contact or a personal distribution list in a messaging store. This document specifies the properties that are unique to Contact and Personal Distribution List objects, and specifies the interdependencies that exist between subsets of these properties. A Contact object can contain subsets of properties that together define a subobject stored on that Contact, such as an e-mail address or a mailing address.

1.4 Relationship to Other Protocols

The Contact Object Protocol Specification is an extension to the Message and Attachment Object Protocol Specification, as specified in [MS-OXCMSG], which defines Message objects.

This protocol also requires an understanding of properties, as specified in [MS-OXPROPS], and **folders**, as specified in [MS-OXCFOLD].

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

A client can use this protocol to represent contact information electronically in a user's **mailbox**.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

None.

2.2 Message Syntax

Contact objects and **Personal Distribution List objects** can be created and modified by clients and servers. Except where noted, this section defines constraints under which both clients and servers operate.

Protocol clients operate on Contact objects and Personal Distribution List objects using the Message and Attachment Object Protocol Specification, as specified in [MS-OXCMSG]. How a protocol server operates on Contact objects and Personal Distribution List objects is implementation-dependent. The results of any such operations are exposed to protocol clients in a manner that is consistent with this protocol.

Unless otherwise specified, a Contact object or a Personal Distribution List object adheres to all **property** constraints specified in [MS-OXPROPS] and all property constraints specified in

[MS-OXCMSG]. A Contact object or Personal Distribution List object MAY also contain other properties <1> <2> <3> as specified in [MS-OXPROPS], but these properties have no impact on this protocol.

2.2.1 Contact Object Properties

The properties specified in section 2.2.1 are specific to Contact objects. They SHOULD only be set on Contact objects. Unless otherwise specified, each of these properties are optional and they only need to be set if there is user data that needs to be stored.

2.2.1.1 Contact Name Properties

The properties specified in section 2.2.1.1 can be set on a Contact object and are used to specify the name of the person represented by the contact.

2.2.1.1.1 PidTagNickname

This optional **PtypString** property specifies the nickname of the contact.

2.2.1.1.2 PidTagGeneration

This optional **PtypString** property specifies the generation suffix of the contact, such as “Jr.”, “Sr.”, “III”.

2.2.1.1.3 PidTagDisplayNamePrefix

This optional **PtypString** property specifies the title of the contact, such as “Mr.”, “Mrs.”.

2.2.1.1.4 PidTagSurname

This optional **PtypString** property specifies the surname (family name) of the contact.

2.2.1.1.5 PidTagMiddleName

This optional **PtypString** property specifies the middle name(s) of the contact.

2.2.1.1.6 PidTagGivenName

This optional **PtypString** property specifies the given name (first name) of the contact.

2.2.1.1.7 PidTagInitials

This optional **PtypString** property specifies the initials of the contact.

2.2.1.1.8 PidTagDisplayName

This optional **PtypString** property specifies the full name of the contact. This property SHOULD be computed by the application according to an implementation-dependent algorithm that uses values of the **PidTagDisplayNamePrefix**, **PidTagGivenName**, **PidTagMiddleName**, **PidTagSurname**, and **PidTagGeneration** properties.

2.2.1.1.9 PidLidYomiFirstName

This optional **PtypString** property specifies the phonetic pronunciation of the contact's given name.

2.2.1.1.10 PidLidYomiLastName

This optional **PtypString** property specifies the phonetic pronunciation of the contact's surname.

2.2.1.1.11 PidLidFileUnder

This **PtypString** property SHOULD be set to the name under which to file this contact when displaying a list of contacts. The application SHOULD treat this property as an empty **PtypString** if it is missing from the Contact object.

2.2.1.1.12 PidLidFileUnderId

This **PtypInteger32** property specifies how to generate and recompute the value of the **PidLidFileUnder** property when other Contact Name properties change. This property SHOULD be set to one of the values in the following table. If this property is missing or set to a value not in the following table, the application can choose its own logic to recompute the value of **PidLidFileUnder** as other contact name properties change.

In the following table, the notation <PropertyName> is used to specify “the value of PropertyName”. For example, if the value of the **PidTagSurname** property is “Smith”, and the value of the **PidTagGivenName** property is “Ben”, then “<PidTagGivenName> <PidTagSurname>” specifies the string “Ben Smith”. In the table, “r” specifies a carriage return character, “n” specifies a line feed character, and <space> represents a space character.

Computing the value of the PidLidFileUnder property based on PidLidFileUnderId property value

Value of the PidLidFileUnderId property	Meaning of the PidLidFileUnder property <4>
0x00000000	Empty PtypString .
0x00003001	“<PidTagDisplayName>”
0x00003A06	“<PidTagGivenName>”
0x00003A11	“<PidTagSurname>”
0x00003A16	“<PidTagCompanyName>”
0x00008017	“<PidTagSurname>, <space><PidTagGivenName> <space><PidTagMiddleName>”

Value of the PidLidFileUnderId property	Meaning of the PidLidFileUnder property <4>
0x00008018	“<PidTagCompanyName>\r\n<PidTagSurname>,<space><PidTagGivenName><space><PidTagMiddleName>”
0x00008019	“<PidTagSurname>,<space><PidTagGivenName><space><PidTagMiddleName>\r\n<PidTagCompanyName>”
0x00008030	“<PidTagSurname><PidTagGivenName><space><PidTagMiddleName>”
0x00008031	“<PidTagSurname><space><PidTagGivenName><space><PidTagMiddleName>”
0x00008032	“<PidTagCompanyName>\r\n<PidTagSurname><PidTagGivenName><space><PidTagMiddleName>”
0x00008033	“<PidTagCompanyName>\r\n<PidTagSurname><space><PidTagGivenName><space><PidTagMiddleName>”
0x00008034	“<PidTagSurname><PidTagGivenName><space><PidTagMiddleName>\r\n<PidTagCompanyName>”
0x00008035	“<PidTagSurname><space><PidTagGivenName><space><PidTagMiddleName>\r\n<PidTagCompanyName>”
0x00008036	“<PidTagSurname><space><PidTagGivenName><space><PidTagMiddleName><space><PidTagGeneration>”
0x00008037	“<PidTagGivenName><space><PidTagMiddleName><space><PidTagSurname><space><PidTagGeneration>”
0x00008038	“<PidTagSurname><PidTagGivenName><space><PidTagMiddleName><space><PidTagGeneration>”
0xffffffff	This value specifies that when displaying the Contact object, the application SHOULD attempt to use the current value of PidLidFileUnder and other contact properties to find a “best match” for PidLidFileUnderId to one of the previous values in this table.
0xffffffffe	This value specifies that when displaying the Contact object, the application SHOULD choose the appropriate default values (according to the language locale) for PidLidFileUnderId and update

Value of the PidLidFileUnderId property	Meaning of the PidLidFileUnder property <4>
	PidLidFileUnder to match the choice.
0xffffffff	PidLidFileUnder is a user-provided PtypString , and SHOULD NOT be changed when another Contact Name property changes.

2.2.1.1.13 PidLidFileUnderList

This optional **PtypMultipleInteger32** property specifies a list of possible values for **PidLidFileUnderId**. Each value in the multi-value property MUST be one of the allowed values for **PidLidFileUnderId** specified in section 2.2.1.1.12.<5>

2.2.1.2 Electronic Address Properties

The Contact object has built-in properties for up to three different e-mail addresses (Email1, Email2, and Email3) and three different fax addresses (Primary Fax, Business Fax, and Home Fax). Each of these electronic addresses is optional, and each consists of a group of related properties associated with it, as specified in the following paragraphs.

Unless otherwise specified, when creating or modifying a Contact object, all properties in a group MUST be set, changed or deleted together at the same time. Specifications of similar properties are grouped together in sections 2.2.1.2.1 through 2.2.1.2.13.

Email1: Refers to the group of properties that define the first e-mail address for a contact. These properties are **PidLidEmail1Displayname**, **PidLidEmail1AddressType**, **PidLidEmail1EmailAddress**, **PidLidEmail1OriginalDisplayName**, and **PidLidEmail1OriginalEntryID**.

Email2: Refers to the group of properties that define the second e-mail address for a contact. These properties are **PidLidEmail2Displayname**, **PidLidEmail2AddressType**, **PidLidEmail2EmailAddress**, **PidLidEmail2OriginalDisplayName**, and **PidLidEmail2OriginalEntryID**.

Email3: Refers to the group of properties that define the third e-mail address for a contact. These properties are **PidLidEmail3Displayname**, **PidLidEmail3AddressType**, **PidLidEmail3EmailAddress**, **PidLidEmail3OriginalDisplayName**, and **PidLidEmail3OriginalEntryID**.

Primary Fax: Refers to the group of properties that define the primary fax address for a contact. These properties are **PidTagPrimaryFaxNumber**, **PidLidFax1AddressType**, **PidLidFax1EmailAddress**, **PidLidFax1OriginalDisplayName**, and **PidLidFax1OriginalEntryID**.

Business Fax: Refers to the group of properties that define the business fax address for a contact. These properties are **PidTagBusinessFaxNumber**, **PidLidFax2AddressType**,

PidLidFax2EmailAddress, PidLidFax2OriginalDisplayName, and PidLidFax2OriginalEntryID.

Home Fax: Refers to the group of properties that define the home fax address for a contact. These properties are **PidTagHomeFaxNumber, PidLidFax3AddressType, PidLidFax3EmailAddress, PidLidFax3OriginalDisplayName, and PidLidFax3OriginalEntryID.**

If any of the e-mail addresses are defined for the contact, then the properties **PidLidAddressBookProviderArrayType** and **PidLidAddressBookProviderEmailList** MUST be defined as well.

For each e-mail address, if one property is defined, then the entire set of properties MUST be defined, and **PidLidAddressBookProviderArrayType** and **PidLidAddressBookProviderEmailList** MUST be defined as well.

If **PidTagPrimaryFaxNumber, PidTagBusinessFaxNumber, or PidTagHomeFaxNumber** is defined, then the **PidLidAddressBookProviderArrayType** and **PidLidAddressBookProviderEmailList** properties SHOULD be defined, as well as the rest of the Primary Fax, Business Fax, or Home Fax properties respectively. <6>

2.2.1.2.1 PidLidEmail1DisplayName, PidLidEmail2DisplayName, PidLidEmail3DisplayName

This **PtypString** property specifies the user-readable display name for the e-mail address.

2.2.1.2.2 PidLidEmail1AddressType, PidLidEmail2AddressType, PidLidEmail3AddressType

This **PtypString** property specifies the address type of the electronic address and, if present, the property value MUST be a valid <7> address type.

2.2.1.2.3 PidLidEmail1EmailAddress, PidLidEmail2EmailAddress, PidLidEmail3EmailAddress

This **PtypString** property specifies the e-mail address of the contact. The value of this property MUST be appropriate <8> for the address type specified for this e-mail address. Address types are specified by the **PidLidEmail1AddressType, PidLidEmail2AddressType, PidLidEmail3AddressType** properties.

2.2.1.2.4 PidLidEmail1OriginalDisplayName, PidLidEmail2OriginalDisplayName, PidLidEmail3OriginalDisplayName

This **PtypString** property SHOULD specify the SMTP e-mail address corresponding to the e-mail address specified in 2.2.1.3.3 for the Contact. In the case that the value of the **PidLidEmail1AddressType, PidLidEmail2AddressType, or PidLidEmail3AddressType** property is "SMTP", the value of respective **PidLidEmail1OriginalDisplayName, PidLidEmail2OriginalDisplayName** or **PidLidEmail3OriginalDisplayName** property SHOULD equal the value of the respective **PidLidEmail1EmailAddress,**

PidLidEmail2EmailAddress, **PidLidEmail3EmailAddress** property. The purpose of this property is to display an alternative user-friendly address that is equivalent to the one in the **PidLidEmail1EmailAddress**, **PidLidEmail2EmailAddress**, or **PidLidEmail3EmailAddress** property.

2.2.1.2.5 PidLidEmail1OriginalEntryID, PidLidEmail2OriginalEntryID, PidLidEmail3OriginalEntryID

This **PtypBinary** property specifies the **EntryId** of the object corresponding to this electronic address. It **MUST** be either a one-off **EntryId** for this electronic address or a valid **Address Book object EntryId**.

2.2.1.2.6 PidTagPrimaryFaxNumber, PidTagBusinessFaxNumber, PidTagHomeFaxNumber

This **PtypString** property specifies the fax number for the contact. The **PtypString** **MUST NOT** be longer than 255 characters. There are no other restrictions on the format of this **PtypString**.

2.2.1.2.7 PidLidFax1AddressType, PidLidFax2AddressType, PidLidFax3AddressType

This **PtypString** property, if present, **MUST** be set to "FAX".

2.2.1.2.8 PidLidFax1EmailAddress, PidLidFax2EmailAddress, PidLidFax3EmailAddress

This **PtypString** property, if present, **SHOULD** contain a user-readable display name, followed by the '@' character, followed by a fax number.

2.2.1.2.9 PidLidFax1OriginalDisplayName, PidLidFax2OriginalDisplayName, PidLidFax3OriginalDisplayName

This **PtypString** property, if present, **MUST** be set to the same value as **PidTagNormalizedSubject**.

2.2.1.2.10 PidLidFax1OriginalEntryID, PidLidFax2OriginalEntryID, PidLidFax3OriginalEntryID

This **PtypBinary** property, if present, **MUST** specify the **one-off EntryId** corresponding to this fax address.

2.2.1.2.11 PidLidEmailList

The value of this **PtypMultipleInteger32** property **MUST** be ignored.<9>

2.2.1.2.12 PidLidAddressBookProviderEmailList

This **PtypMultipleInteger32** property specifies which electronic address properties are set on the Contact object. Each **PtypInteger32** value in this property **MUST** be unique in the property and **MUST** be set to one of the values in the following table. If this property is set,

then **PidLidAddressBookProviderArrayType** MUST also be set. These two properties MUST be kept synchronized with each other. <10> For example, if one of the values in **PidLidAddressBookProviderEmailList** is 0x00000000, then **PidLidAddressBookProviderArrayType** MUST have the bit 0x00000001 set.

Value	Meaning
0x00000000	Email1 is defined for the contact.
0x00000001	Email2 is defined for the contact.
0x00000002	Email3 is defined for the contact.
0x00000003	Business fax is defined for the contact.
0x00000004	Home fax is defined for the contact.
0x00000005	Primary fax is defined for the contact.

2.2.1.2.13 PidLidAddressBookProviderArrayType

This **PtypInteger32** property specifies the state of the contact's electronic addresses and represents a set of bit-flags. The value of the **PidLidAddressBookProviderArrayType** property MUST be a combination of flags that specify the state of the Contact object. Individual flags are specified in the following table. If this property is set, then **PidLidAddressBookProviderEmailList** MUST be set as well. These two properties MUST be kept in sync with each other. For example, if this property has the bit 0x00000001 set, then one of the values of **PidLidAddressBookProviderEmailList** MUST be 0x00000000.

Bit	Meaning
0x00000001	Email1 is defined for the contact
0x00000002	Email2 is defined for the contact.
0x00000004	Email3 is defined for the contact.
0x00000008	Business fax is defined for the contact.
0x00000010	Home fax is defined for the contact.
0x00000020	Primary fax is defined for the contact.

2.2.1.3 Physical Address Properties

The Contact object provides built-in support for three physical addresses: the Home Address, the Work Address, and the Other Address. One of the addresses can be marked as the Mailing Address. Each of these physical addresses is optional, and each consists of a group of related properties associated with it, as specified in this section.

When creating or modifying a Contact object, all properties in a group **MUST** be set, changed, or deleted together at the same time. Specifications of similar properties are grouped together in sections 2.2.1.3.1 through 2.2.1.3.9.

Unless otherwise specified, for each physical address, if one property is set, then all the properties for that address **MUST** also be set.

The Home Address refers to the address specified by the properties **PidTagHomeAddressStreet**, **PidTagHomeAddressCity**, **PidTagHomeAddressStateOrProvince**, **PidTagHomeAddressPostalCode**, **PidTagHomeAddressCountry**, **PidLidHomeAddressCountryCode**, and **PidLidHomeAddress**.

The Work Address refers to the address specified by the properties **PidLidWorkAddressStreet**, **PidLidWorkAddressCity**, **PidLidWorkAddressState**, **PidLidWorkAddressPostalCode**, **PidLidWorkAddressCountry**, **PidLidWorkAddressCountryCode**, and **PidLidWorkAddress**.

The Other Address refers to the address specified by the properties **PidTagOtherAddressStreet**, **PidTagOtherAddressCity**, **PidTagOtherAddressStateOrProvince**, **PidTagOtherAddressPostalCode**, **PidTagOtherAddressCountry**, **PidLidOtherAddressCountryCode**, and **PidLidOtherAddress**.

The Mailing Address refers to the address specified by the properties **PidTagStreetAddress**, **PidTagLocality**, **PidTagStateOrProvince**, **PidTagPostalCode**, **PidTagCountry**, **PidLidAddressCountryCode**, and **PidTagPostalAddress**.

2.2.1.3.1 **PidLidWorkAddressStreet**, **PidTagHomeAddressStreet**, **PidTagOtherAddressStreet**, **PidTagStreetAddress**

This **PtypString** property specifies the street part of the contact's work, home, other, or mailing address. <11>

2.2.1.3.2 **PidLidWorkAddressCity**, **PidTagHomeAddressCity**, **PidTagOtherAddressCity**, **PidTagLocality**

This **PtypString** property specifies the city or locality part of the contact's work, home, other or mailing address.

2.2.1.3.3 **PidLidWorkAddressState**, **PidTagHomeAddressStateOrProvince**, **PidTagOtherAddressStateOrProvince**, **PidTagStateOrProvince**

This **PtypString** property specifies the state or province part of the contact's work, home, other or mailing address.

2.2.1.3.4 PidLidWorkAddressPostalCode, PidTagHomeAddressPostalCode, PidTagOtherAddressPostalCode, PidTagPostalCode

This **PtypString** property specifies the postal code (ZIP Code) part of the contact's work, home, other or mailing address.

2.2.1.3.5 PidLidWorkAddressCountry, PidTagHomeAddressCountry, PidTagOtherAddressCountry, PidTagCountry

This **PtypString** property specifies the country or region part of the contact's work, home, other or mailing address.

2.2.1.3.6 PidLidWorkAddressCountryCode, PidLidHomeAddressCountryCode, PidLidOtherAddressCountryCode, PidLidAddressCountryCode

This **PtypString** property specifies the country code part of the contact's work, home, other or mailing address.

2.2.1.3.7 PidLidWorkAddressPostOfficeBox, PidTagHomeAddressPostOfficeBox, PidTagOtherAddressPostOfficeBox, PidTagPostOfficeBox

This **PtypString** property specifies the post office box part of the contact's work, home, other or mailing address.<12>

2.2.1.3.8 PidLidWorkAddress, PidLidHomeAddress, PidLidOtherAddress, PidTagPostalAddress

This **PtypString** property specifies the complete address of the contact's work, home, other or mailing address. This property SHOULD be a combination of other physical address properties and is based on client locale.

2.2.1.3.9 PidLidPostalAddressId

This optional **PtypInteger32** property specifies which physical address is the mailing address for this contact. If present, the property MUST have one of the values in the following table. If not set, the application SHOULD assume that the value is 0x00000000.

Value	Description
0x00000000	No address is selected as the mailing address. PidTagStreetAddress, PidTagLocality, PidTagStateOrProvince, PidTagPostalCode, PidTagCountry, PidLidAddressCountryCode, and PidTagPostalAddress MUST all be not set.
0x00000001	The Home Address is the mailing address.

Value	Description
	The values of the PidTagStreetAddress , PidTagLocality , PidTagStateOrProvince , PidTagPostalCode , PidTagPostOfficeBox , PidTagCountry , PidLidAddressCountryCode , and PidTagPostalAddress properties MUST be equal to the values of the PidTagHomeAddressStreet , PidTagHomeAddressCity , PidTagHomeAddressStateOrProvince , PidTagHomeAddressPostalCode , PidTagHomeAddressPostOfficeBox , PidTagHomeAddressCountry , PidLidHomeAddressCountryCode , and PidLidHomeAddress properties, respectively.
0x00000002	The Work Address is the mailing address. The values of the PidTagStreetAddress , PidTagLocality , PidTagStateOrProvince , PidTagPostalCode , PidTagPostOfficeBox , PidTagCountry , PidLidAddressCountryCode , and PidTagPostalAddress properties MUST be equal to the values of the PidLidWorkAddressStreet , PidLidWorkAddressCity , PidLidWorkAddressState , PidLidWorkAddressPostalCode , PidLidWorkAddressPostOfficeBox , PidLidWorkAddressCountry , PidLidWorkAddressCountryCode , and PidLidWorkAddress properties, respectively.
0x00000003	The Other Address is the mailing address. The values of the PidTagStreetAddress , PidTagLocality , PidTagStateOrProvince , PidTagPostalCode , PidTagPostOfficeBox , PidTagCountry , PidLidAddressCountryCode , and PidTagPostalAddress properties MUST be equal to the values of the PidTagOtherAddressStreet , PidTagOtherAddressCity , PidTagOtherAddressStateOrProvince , PidTagOtherAddressPostalCode , PidTagOtherAddressPostOfficeBox , PidTagOtherAddressCountry , PidLidOtherAddressCountryCode , and PidLidOtherAddress properties, respectively.

2.2.1.4 Telephone Properties

The following properties all specify optional telephone numbers for the contact. If present, each property MUST NOT exceed a length of 255 characters, excluding the NULL character. Each of the properties in this section are optional.

2.2.1.4.1 PidTagPagerTelephoneNumber

This **PtypString** property specifies the pager telephone number.

2.2.1.4.2 PidTagCallbackTelephoneNumber

This **PtypString** property specifies the callback telephone number.

2.2.1.4.3 PidTagBusinessTelephoneNumber

This **PtypString** property specifies the business telephone number.

2.2.1.4.4 PidTagHomeTelephoneNumber

This **PtypString** property specifies the home telephone number.

2.2.1.4.5 PidTagPrimaryTelephoneNumber

This **PtypString** property specifies the primary telephone number.

2.2.1.4.6 PidTagBusiness2TelephoneNumber

This **PtypString** property specifies the second business telephone number.

2.2.1.4.7 PidTagMobileTelephoneNumber

This **PtypString** property specifies the mobile telephone number.

2.2.1.4.8 PidTagRadioTelephoneNumber

This **PtypString** property specifies the radio telephone number.

2.2.1.4.9 PidTagCarTelephoneNumber

This **PtypString** property specifies the car telephone number.

2.2.1.4.10 PidTagOtherTelephoneNumber

This **PtypString** property specifies an alternate telephone number.

2.2.1.4.11 PidTagAssistantTelephoneNumber

This **PtypString** property specifies the telephone number of the contact's assistant.

2.2.1.4.12 PidTagHome2TelephoneNumber

This **PtypString** property specifies a second home telephone number.

2.2.1.4.13 PidTagTtyTddPhoneNumber

This **PtypString** property specifies the telephone number for the contact's text telephone (TTY) or telecommunication device for the deaf (TDD).

2.2.1.4.14 PidTagCompanyMainPhoneNumber

This **PtypString** property specifies the company phone number.

2.2.1.4.15 PidTagTelexNumber

This **PtypString** property specifies the telex number.

2.2.1.4.16 PidTagIsdnNumber

This **PtypString** property specifies the integrated services digital network (ISDN) number.

2.2.1.5 Event Properties

There are two events associated with a contact, a birthday and an anniversary. Each event is defined by two properties: a time property and an object **EntryId** property. If one of the two properties is set for an event, the other **MUST** also be set. If either event is defined for a contact, then **PidLidReferenceEntryId** **SHOULD** be set and its value **SHOULD** be equal to the value of **PidTagEntryId**.

2.2.1.5.1 PidTagBirthday

If present, this **PtypFloatingTime** property specifies the birthday of the contact, at 12:00 AM UTC.

2.2.1.5.2 PidLidBirthdayEventEntryId

If present, this **PtypBinary** property specifies the object **EntryId** of the optional appointment object, as specified by [MS-OXOCAL], that represents the contact's birthday. The Appointment object specified by **PidLidBirthdayEventEntryId** **MUST** be linked to this contact using **PidLidContactLinkEntry**, **PidLidContactSearchKey**, **PidLidContactLinkName**, as specified in [MS-OXCMSG].

2.2.1.5.3 PidTagWeddingAnniversary

If present, this **PtypFloatingTime** property specifies the wedding anniversary of the contact, at 12:00 AM UTC.

2.2.1.5.4 PidLidAnniversaryEventEntryId

If present, this **PtypBinary** property specifies the object **EntryId** of the Appointment object, as specified by [MS-OXOCAL], that represents the contact's anniversary. The Appointment object specified by **PidLidAnniversaryEventEntryId** **MUST** be linked to this contact using **PidLidContactLinkEntry**, **PidLidContactSearchKey**, **PidLidContactLinkName**, as specified in [MS-OXCMSG].

2.2.1.6 Professional Properties

The following properties are used to store professional details for the person represented by the contact.

2.2.1.6.1 PidTagTitle

This optional **PtypString** property specifies the job title of the contact.

2.2.1.6.2 PidTagCompanyName

This optional **PtypString** property specifies the company that employs the contact.

2.2.1.6.3 PidTagDepartmentName

This optional **PtypString** property specifies the name of the department that the contact is a part of.

2.2.1.6.4 PidLidDepartment

This optional **PtypString** property MUST be ignored.<13>

2.2.1.6.5 PidTagOfficeLocation

This optional **PtypString** property specifies the location of the office that the contact works in.

2.2.1.6.6 PidTagManagerName

This optional **PtypString** property specifies the name of the contact's manager.

2.2.1.6.7 PidTagAssistant

This optional **PtypString** property specifies the name of the contact's assistant.

2.2.1.6.8 PidLidYomiCompanyName

This optional **PtypString** property specifies the phonetic pronunciation of the contact's company name.

2.2.1.6.9 PidTagProfession

This optional **PtypString** property specifies the profession of the contact.

2.2.1.7 Business Card Properties

The properties specified in this section can be used to customize the display of contact information in business card format.<14> The business card format is a collection of information about how to display contact data to the user.

2.2.1.7.1 PidLidBusinessCardDisplayDefinition

PidLidBusinessCardDisplayDefinition is an optional **PtypBinary** property that contains user customization details for displaying a contact as a business card. The layout of a business card can be represented as an image and a number of text fields. The image can be either the contact photo, specified in section 2.2.1.8, or the card picture, specified in section 2.2.1.7.2. Text fields consist of a value from another **PtypString** property set on the Contact object and an optional customized label string provided by the user.

The following sections specify the format of the **PidLidBusinessCardDisplayDefinition** property. Note that multi-byte values are stored in **little-endian** format in the buffer.

2.2.1.7.1.1 PidLidBusinessCardDisplayDefinition Buffer Format

The following table lists the buffer format of the **PidLidBusinessCardDisplayDefinition** property.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
MajorVersion										MinorVersion										TemplateID										CountOfFields									
FieldInfoSize										ExtraInfoSize										ImageAlignment										ImageSource									
BackgroundColor																																							
ImageArea										Reserved																													
...										FieldInfo1 (variable)																													
...																																							
FieldInfoN (variable)																																							
ExtraInfo (variable)																																							

MajorVersion (1 byte): An 8-bit value that specifies the major version number. See section 2.2.1.7.1.1.1.

MinorVersion (1 byte): An 8-bit value that specifies the minor version number. See section 2.2.1.7.1.1.1.

TemplateID (1 byte): An 8-bit value that specifies the display template to use. See section 2.2.1.7.1.1.2.

CountOfFields (1 byte): An 8-bit value that specifies the count of **FieldBuffer** structures in the **PidLidBusinessCardDisplayDefinition**.

FieldInfoSize (1 byte): An 8-bit value that specifies the size, in bytes, of **FieldInfo** structures, as specified in section 2.2.1.7.1.2. This value **MUST** be greater or equal to 0x10.

ExtraInfoSize (1 byte): An 8-bit value that specifies the size, in bytes, of any additional data provided in the **ExtraInfo** BYTE Array, specified in section 2.2.1.7.1.3.

ImageAlignment (1 byte): An 8-bit value that specifies the image alignment in the image area, as specified in section 2.2.1.7.1.1.3.

ImageSource (1 byte): An 8-bit value that specifies the image source, as specified in section 2.2.1.7.1.1.4.

BackgroundColor (4 bytes): A **PtypInteger32** that specifies the background business card color, as specified in section 2.2.1.7.1.1.5.

ImageArea (1 byte): An 8-bit value that specifies the percent of space on the card the image will take up, as specified in section 2.2.1.7.1.1.6.

Reserved (4 bytes): MUST be set to 0x00000000.

FieldInfo1 (variable): A structure value that contains field information, as specified in section 2.2.1.7.1.2.

FieldInfoN (variable): A structure that contains field information, as specified in section 2.2.1.7.1.2. The number of **FieldInfo** structures included in the buffer is equal to **CountOfFields**.

ExtraInfo (ExtraInfoSize): A byte array that specifies additional information, as specified in 2.2.1.7.1.3.

2.2.1.7.1.1.1 MajorVersion and MinorVersion

The value of the **MajorVersion** field MUST be 0x03 or greater. The value of the **MinorVersion** MAY be any value. A user agent implementing this protocol SHOULD set the value of **MajorVersion** to 0x03 and SHOULD set the value of **MinorVersion** to 0x00.<15>

2.2.1.7.1.1.2 TemplateID

This field represents the business card layout type. The value of this field MUST be set to one of the following values:

Value	Description
0x00	The image will be left aligned, stretching the full height of the card vertically; text fields will appear to the right of the image.
0x01	The image will be right aligned, stretching the full height of the card vertically; text fields will appear to the left of the image.
0x02	The image will be aligned to the top, stretched the full width of the card horizontally; text fields will appear under the image.
0x03	The image will be aligned to the bottom, stretched the full width of the card horizontally; text fields will appear above the image
0x04	No image is included in the card, only text fields are included. PidlBusinessCardCardPicture SHOULD NOT be set on the Contact object in this case.
0x05	The image will be used as a background for the card, stretching the full height and width of the card. Text fields are displayed on top of the image.

2.2.1.7.1.1.3 ImageAlignment

This field indicates how the image is aligned in the image area. The value of this field is ignored for text only cards (i.e. when the value of **TemplateID** is 0x04). The value of this field **MUST** have one of the following values:

Value	Description
0x00	Image is stretched to fit
0x01	Top left
0x02	Align top center
0x03	Align top right
0x04	Align middle left
0x05	Align middle center
0x06	Align middle right
0x07	Align bottom left
0x08	Align bottom center
0x09	Align bottom right

2.2.1.7.1.1.4 ImageSource

The business card can display up to one image on the card. That image can be obtained from either the contact photo, as specified in section 2.2.1.8, or the card picture, as specified in section 2.2.1.7.2. If the value of the **ImageSource** field is 0x00 the contact photo **SHOULD** be used, otherwise the card picture property **SHOULD** be used. If the value of this field is 0x00, the **PidLidBusinessCardCardPicture** property **SHOULD NOT** exist on the Contact object. The value of this field is ignored for text only cards (when the value of **TemplateID** is 0x04).

2.2.1.7.1.1.5 BackgroundColor

A **PtypInteger32** value representing the color of the card background, expressed as 0x00BBGGRR, where the high byte is 0x00, the next highest byte identifies the blue intensity value, the next highest byte identifies the green intensity value, and the lowest byte identifies the red intensity value.

2.2.1.7.1.1.6 ImageArea

This field indicates the percentage of space on the card on which to display the image. The value of this field **SHOULD** be between 0x04 and 0x32 (representing 4% and respectively

50%). The value of this field is ignored for text only cards and background image cards (when the value of **TemplateID** is 0x04 or 0x05).

2.2.1.7.1.2 FieldInfo Buffer Format

The following table lists the buffer format of the **FieldInfo** structure.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
TextPropertyID																TextFormat								LabelFormat							
FontSize				Reserved								LabelOffset																			
ValueFontColor																															
LabelFontColor																															

TextPropertyID (2 bytes): A 16-bit value that specifies the property ID of the field, as specified in section 2.2.1.7.1.2.1.

TextFormat (1 byte): An 8-bit value that specifies the text decoration and alignment information, as specified in section 2.2.1.7.1.2.2.

LabelFormat (1 byte): An 8-bit value that specifies the label information, as specified in section 2.2.1.7.1.2.3.

FontSize (1 byte): An 8-bit value that specifies the font size in **points**.

Reserved (1 byte): MUST be set to 0x00.

LabelOffset (2 bytes): A **PtypInteger16** value that specifies the byte offset into extra byte information, as specified in section 2.2.1.7.1.2.5.

ValueFontColor (4 bytes): A **PtypInteger32** value that specifies the color reference code for the value font color.

LabelFontColor (4 bytes): A **PtypInteger32** value that specifies the color reference code for the label font color.

2.2.1.7.1.2.1 TextPropertyID

The value of this field MUST be either 0x0000, representing an empty field, or the property ID of one of the properties from the following list. Note that all properties in the list are **PtypString** properties.

Allowed Properties

PidTagDisplayName

PidTagTitle

PidTagDepartmentName
PidTagCompanyName
PidTagBusinessTelephoneNumber
PidTagBusiness2TelephoneNumber
PidTagBusinessFaxNumber
PidTagCompanyMainPhoneNumber
PidTagHomeTelephoneNumber
PidTagHome2TelephoneNumber
PidTagHomeFaxNumber
PidTagMobileTelephoneNumber
PidTagAssistantTelephoneNumber
PidTagOtherTelephoneNumber
PidTagTtydddPhoneNumber
PidTagPrimaryTelephoneNumber
PidTagPrimaryFaxNumber
PidTagPagerTelephoneNumber
PidLidWorkAddress
PidLidHomeAddress
PidLidOtherAddress
PidLidInstantMessaging
PidTagBusinessHomePage
PidTagPersonalHomePage
PidLidContactUserField1
PidLidContactUserField2
PidLidContactUserField3
PidLidContactUserField4
PidLidEmail1OriginalDisplayName
PidLidEmail2OriginalDisplayName
PidLidEmail3OriginalDisplayName

2.2.1.7.1.2.2 TextFormat

This byte value contains bit flags that indicate alignment and font formatting for the text value of the field. If none of the bits defined in the following table are set, the field text is displayed single line, left-aligned. The **Right align** and **Center align** bits **MUST** be mutually exclusive.

0	1	2	3	4	5	6	7
MUST be 0		Center align	Right align	Underline	Italic	Bold	Multiline

2.2.1.7.1.2.3 LabelFormat

This byte value contains bit flags that indicate the presence and alignment of a custom label associated with the field text. If none of the bits defined in the following table are set, the field has no label. The **Label to the right** and **Label to the left** bits **MUST** be mutually exclusive.

0	1	2	3	4	5	6	7
MUST be 0					Use right-to-left reading order	Label to the left	Label to the right

2.2.1.7.1.2.4 FontSize

FontSize **MUST** be a value between 0x03 and 0x20 (representing 3 and 32) indicating the font size, in **points**, to be used by this field text. The value of this field **MUST** be set to 0x00 if the field represents an empty line.

2.2.1.7.1.2.5 LabelOffset

LabelOffset **MUST** be set to the byte offset into the **ExtraInfo** buffer pointing at the start of the label string. If the field does not have a label, the value of **LabelOffset** **MUST** be 0xFFFE. All label strings **MUST** be stored as **Unicode**, null-terminated **PtypStrings** in the **ExtraInfo** buffer. Each label **SHOULD** be limited to 16 Unicode characters, including the null terminator. The value of **LabelOffset** **MUST** be less than the value of **ExtraInfoSize**, which is the total size of the **ExtraInfo** buffer.

2.2.1.7.1.2.6 ValueFontColor

ValueFontColor is a color reference code indicating the font color of the value. (0x00BBGRR), where the high byte is 0x00, the next highest byte identifies the blue intensity value, the next highest byte identifies the green intensity value, and the lowest byte identifies the red intensity value.

2.2.1.7.1.2.7 LabelFontColor

LabelFontColor is a color reference code indicating the font color of the label (0x00BBGRR), where the high byte is 0x00, the next highest byte identifies the blue intensity value, the next highest byte identifies the green intensity value, and the lowest byte identifies the red intensity value.

2.2.1.7.1.3 ExtraInfo Buffer Format

This byte array buffer contains a set of Unicode **PtypString** values of labels that have been customized by the user. The labels **MUST** be stored as Unicode **PtypStrings**, each ending in a NULL terminator. Each of these **PtypStrings** **SHOULD** be referenced by a **LabelOffset** fields in one or more **FieldInfo** structures, as specified in section 2.2.2.6.2.2. The total size, in bytes, of the **ExtraInfo** field **MUST** be specified by the value of **ExtraInfoSize** field.

2.2.1.7.2 PidLidBusinessCardCardPicture

PtypBinary property that contains the image to be used on a business card, whose value **MUST** be either a portable network graphics (PNG) or joint photographic experts group (JPEG) stream. This property **SHOULD** be used in conjunction with **PidLidBusinessCardDisplayDefinition** as follows: **PidLidBusinessCardCardPicture** **SHOULD NOT** be present on a Contact object if **PidLidBusinessCardDisplayDefinition** is not present. This property also **SHOULD NOT** be present if the data in **PidLidBusinessCardDisplayDefinition** (as specified in section 2.2.1.7.1.1) do not require a card image.

2.2.1.7.3 PidLidContactUserField1, PidLidContactUserField2, PidLidContactUserField3, PidLidContactUserField4

These four optional **PtypString** properties can be used to add custom text to a business card representation of a Contact object. These properties contain text that is unrelated to any other contact-specific property.

2.2.1.8 Contact Photo Properties

The property and attachment specified in this section are optional and represent an optional photo associated with the contact.<16>

2.2.1.8.1 PidLidHasPicture

If this **PtypBoolean** property exists and is set to TRUE, then the attachment specified in section 2.2.1.8.2 **SHOULD** exist and the application **SHOULD** use it as the contact photo.

2.2.1.8.2 Contact Photo Attachment

The contact photo attachment is a picture attached to the Contact object. For more details about Message object attachments, see [MS-OXCMSG]. Additionally, the following properties **MUST** be set on the attachment object, as specified in [MS-OXCMSG]:

- The value of the **PtypBoolean PidTagAttachmentContactPhoto** property MUST be set to TRUE. There SHOULD <17> only be one attachment with **PidTagAttachmentContactPhoto** set to TRUE on a given Contact object.
- The value of the **PtypBinary PidTagAttachDataBin** property (i.e. the contents of the attachment) MUST be a stream in JPEG format.
- The value of the **PtypString PidTagAttachExtension PtypString** property MUST be set to “.jpg”.
- The value of the **PidTagDisplayName** and **PidTagAttachFilename PtypString** properties MUST be set to “ContactPicture.jpg”.

2.2.1.9 Other Contact Properties

2.2.1.9.1 PidLidReferenceEntryId

If present, this **PtypBinary** property SHOULD be equal to the value of the EntryId of the Contact object.<18>

2.2.1.9.2 PidTagHobbies

This optional **PtypString** property specifies the hobbies of the contact.

2.2.1.9.3 PidTagSpouseName

This optional **PtypString** property specifies the name of the contact’s spouse.

2.2.1.9.4 PidTagLanguage

This optional **PtypString** property specifies the language that the contact uses.

2.2.1.9.5 PidTagLocation

This optional **PtypString** specifies the location of the contact. For example, this could be the building and office number of the contact.

2.2.1.9.6 PidLidInstantMessaging

This optional **PtypString** property specifies the contact’s instant messaging address.

2.2.1.9.7 PidTagOrganizationalIdNumber

This optional **PtypString** property specifies an organizational ID number of the contact, for example an employee ID number.

2.2.1.9.8 PidTagCustomerId

This optional **PtypString** property specifies the contact’s customer ID number.

2.2.1.9.9 PidTagGovernmentIdNumber

This optional **PtypString** property specifies the contact's government ID number.

2.2.1.9.10 PidLidFreeBusyLocation

This optional **PtypString** property specifies a URL path from which a client can retrieve free/busy information for this contact as an iCal file, as specified by [MS-OXCICAL]

2.2.1.9.11 PidTagAccount

This optional **PtypString** property specifies the account name of the contact. <19>

2.2.1.9.12 PidLidHtml

This optional **PtypString** property specifies the contact's business web page URL. The value of this property, if present, SHOULD be the same as the value of **PidTagBusinessHomePage**.

2.2.1.9.13 PidTagPersonalHomePage

This optional **PtypString** property specifies the contact's personal web page URL.

2.2.1.9.14 PidTagBusinessHomePage

This optional **PtypString** property specifies the contact's business web page URL. The value of this property, if present, SHOULD be the same as the value of **PidLidHtml**.

2.2.1.9.15 PidTagFtpSite

This optional **PtypString** property specifies the contact's File Transfer Protocol (FTP) URL. FTP is a protocol used to transfer data, as specified in [RFC959].

2.2.1.9.16 PidTagComputerNetworkName

This optional **PtypString** property specifies the name of the network that the contact's computer is connected to. <20>

2.2.1.9.17 PidTagChildrensNames

This optional **PtypMultipleString** property specifies the names of the contact's children.

2.2.1.9.18 PidLidContactCharSet

This optional unsigned **PtypInteger32** property specifies the character set <21> used for this Contact object. Applications can use this property to aid in generating a character-set dependent list of choices for the properties **PidLidFileUnder**, **PidLidFileUnderList**, and **PidLidFileUnderId**. If the value of the property is 0x00000000 or 0x00000001, applications SHOULD treat the property as not being set.

2.2.1.9.19 PidLidAutoLog

This optional **PtypBoolean** property specifies to the application whether or not to create a Journal object for each action associated with this Contact object.

2.2.1.9.20 PidTagGender

This optional **PtypInteger16** property specifies the gender of the contact. If present, the property MUST be one of the following values.

Value	Meaning
0x0000	The contact's gender is unspecified
0x0001	The contact is female
0x0002	The contact is male

2.2.1.9.21 PidTagReferredByName

This optional **PtypString** property specifies the name of a person who referred this contact to the user.

2.2.1.9.22 PidLidContactItemData

This optional **PtypMultipleInteger32** property can be used to help display the contact information. If present, it MUST have 6 entries, each corresponding to a visible field in the application's user interface.

The meaning of each entry in the **PtypMultipleInteger32** property is defined in the following table.

One-based index into the multi-valued property	The value MUST be one of the following	Meaning
1	0x00000001	The application SHOULD display the contact's home address .
	0x00000002 or 0x00000000	The application SHOULD display the contact's work address.
	0x00000003	The application SHOULD display the contact's other address.
2	0x00008080	The application SHOULD display Email 1

One-based index into the multi-valued property	The value MUST be one of the following	Meaning
	0x00008090	The application SHOULD display Email 2
	0x000080A0	The application SHOULD display Email 3
3,4,5,6	PropertyID of any of the Telephone Properties specified in section 2.2.1.4 or of any of the Fax Numbers specified in section 2.2.1.2.6.	The application SHOULD display the corresponding property

2.2.1.9.23 PidLidUserX509Certificate

This optional **PtypMultipleBinary** property specifies a list of certificates for the contact. The format and semantics of this property is the same as specified in [MS-OXOABK] for the PidTagUserX509Certificate.

2.2.1.10 Additional Property Constraints

This protocol specifies additional constraints on the following properties in addition to what is specified in [MS-OXCMSG].

2.2.1.10.1 PidTagNormalizedSubject

This optional **PtypString** property specifies a combination of the full name and company name of the contact. This property SHOULD be computed by the application according to an implementation-dependent algorithm that uses values of the **PidTagGivenName**, **PidTagMiddleName**, **PidTagSurname**, **PidTagGeneration**, **PidTagCompanyName** properties.<22>

2.2.1.10.2 PidTagMessageClass

A Contact object MUST have this **PtypString** property set to “IPM.Contact”, or a **PtypString** prefixed with “IPM.Contact.”

2.2.2 Personal Distribution List Properties

The properties specified in this section are specific to Personal Distribution List objects. They SHOULD only be set on Personal Distribution List objects.

2.2.2.1 Personal Distribution List Name Properties

The following properties are used to display the name of the Personal Distribution List object.

2.2.2.1.1 PidTagDisplayName

This **PtypString** property specifies the user-visible name of the personal distribution list.

2.2.2.1.2 PidLidDistributionListName

This **PtypString** property specifies the name of the personal distribution list. The value of this property SHOULD <23> be the same as the value of the **PidTagDisplayName** property.

2.2.2.1.3 PidLidFileUnder

The value of this **PtypString** property MUST be the same as the value of the **PidTagDisplayName** property.

2.2.2.1.4 PidLidFileUnderId

This optional **PtypInteger32** property SHOULD <24> be set to 0xffffffff if present.

2.2.2.2 Personal Distribution List Member Properties

2.2.2.2.1 PidLidDistributionListMembers

This **PtypMultipleBinary** property specifies the list of EntryIds of the objects corresponding to the members of the personal distribution list. Members of the personal distribution list can be other Personal Distribution List, electronic addresses contained in a Contact, Global Address List users or distribution lists, or one-off e-mail addresses. The format of each EntryId MUST be either a one-off EntryId (as specified in [MS-OXCDATA]) or a Wrapped EntryId, as specified in section 2.2.2.2.5.

When setting this property, the client or the server MUST ensure its total size is less than 15000 bytes.

2.2.2.2.2 PidLidDistributionListOneOffMembers

This **PtypMultipleBinary** property specifies the list of one-off EntryIds corresponding to the members of the personal distribution list. These one-off EntryIds encapsulate display names and e-mail addresses of the Personal Distribution List members.

If the client or the server set this property <25>, the property MUST be synchronized with **PidLidDistributionListMembers**: for each entry in the **PidLidDistributionListOneOffMembers** property, there MUST be an entry in the same position in the **PidLidDistributionListMembers** property.

When setting this property, the client or the server MUST ensure that its total size is less than 15000 bytes in size.

2.2.2.2.3 PidLidDistributionListChecksum

This **PtypInteger32** property specifies the 32-bit cyclic redundancy check (CRC-32) polynomial checksum, as specified in [ISO/IEC 8802-3:2000], calculated on the value of **PidLidDistributionListMembers**, as specified in section 3.1.4.2.2.4. The value of this

property can be used to detect when the **PidLidDistributionListMembers** property was updated without updating the other Personal Distribution List Member properties (by computing the CRC-32 on the existing value of **PidLidDistributionListMembers** and comparing it with the value of the **PidLidDistributionListChecksum** property).<26>

2.2.2.2.4 Wrapped EntryId Format

When used as members of a Personal Distribution List, Address Book EntryIds can be “wrapped” inside a Wrapped EntryId. The format of the Wrapped EntryId is specified in the following table:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Flags																															
Provider UID																															
...																															
...																															
...																															
Type				Embedded EntryId (variable)																											
...																															

Flags (4 bytes): MUST be 0x00000000.

Provider UID (16 bytes): MUST be %xC0.91.AD.D3.51.9D.CF.11.A4.A9.00.AA.00.47.FA.A4

Type (1 byte): This bit-field specifies how the next field is interpreted, and MUST be a combination of flags from the following table:

Bit Mask	Description
0x0F	<p>The lower 4 bits of the Type field are interpreted as an unsigned integer value that specifies what type of EntryId is embedded in this Wrapped EntryId. These bits MUST be set to one of the following values:</p> <ul style="list-style-type: none"> • A value of 0 designates a one-off EntryId; the Embedded EntryId field MUST be a one-off EntryId as specified in [MS-OXCDATA]. • A value of 3 designates a Contact object EntryId; the Embedded EntryId field MUST be the Message EntryId (as specified in [MS-OXCDATA]) of a contact object. • A value of 4 designates a Personal Distribution List object EntryId ; the Embedded EntryId field MUST be the Message EntryId (as specified in [MS-OXCDATA]) of a personal distribution list object. • A value of 5 designates a Global Address List mail user EntryId ; the

Bit Mask	Description
	<p>Embedded EntryId field MUST be the Global Address List EntryId (as specified in [MS-OXCADATA]) of a user in the Global Address List.</p> <ul style="list-style-type: none"> • A value of 6 designates a Global Address List distribution list EntryId; the Embedded EntryId field MUST be the Global Address List EntryId (as specified in [MS-OXCADATA]) of a distribution list in the Global Address List.
0x70	<p>The next 3 bits of the Type field are interpreted as an unsigned integer value. If the embedded EntryId is a one-off EntryId, this value MUST be 0. Otherwise, if the embedded EntryId is not a Contact object EntryId, this value MUST be 3.</p> <p>If the embedded EntryId is a Contact object EntryId, this unsigned integer MUST have one of the following values:</p> <ul style="list-style-type: none"> • 0 (denoting a Business Fax electronic address) • 1 (denoting a Home Fax electronic address) • 2 (denoting a Primary Fax electronic address) • 4 (denoting an E-mail1 address) • 5 (denoting an E-mail2 address) • 6 (denoting an Email3 address) <p>Note that this value MUST NOT be set to 7 in any case.</p>
0x80	<p>If the embedded EntryId is a one-off EntryId, this bit MUST NOT be set. Otherwise, this bit MUST be set.</p>

Embedded EntryId (variable): This field represents an EntryId that MUST be interpreted according to the Type field, more specifically according to the value contained in the lower 4 bits of the Type field, as specified in the previous table.

2.2.2.3 Other Personal Distribution List Properties

2.2.2.3.1 PidLidAddressBookProviderArrayType

This **PtypInteger32** property SHOULD <27> be set to 0x00000000.

2.2.2.4 Additional Property Constraints

This protocol specifies additional constraints on the following properties beyond what is specified in [MS-OXCMSG].

2.2.2.4.1 PidTagNormalizedSubject

The value of this **PtypString** property MUST be the same as the value of the **PidTagDisplayName** property.

2.2.2.4.2 PidTagMessageClass

A Personal Distribution List object MUST have this **PtypString** property set to “IPM.DistList” or a **PtypString** prefixed with “IPM.DistList.”

2.2.3 Contact Folder Syntax

Contact folders are folders, as specified in [MS-OXCFOLD], that have the value of the **PidTagContainerClass PtypString** property set to “IPF.Contact”, or a **PtypString** prefixed with “IPF.Contact.”. In all other respects they conform to the folder syntax and functionality specified by [MS-OXCFOLD].

Contact objects and Personal Distribution List objects SHOULD be created in contact folders.

3 Protocol Details

3.1 Client and Server Details

The client and server roles create and manipulate contacts and personal distribution lists, and otherwise operate within their roles, as specified in [MS-OXCMSG].

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation could maintain to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

3.1.1.1 Contact Object

Contact objects extend the Message objects specified in [MS-OXCMSG] and define additional contact-specific properties. These properties are either stand-alone properties, or they are used in groups to define conceptual subobjects. Each of these subobjects are defined in sections 3.1.1.1.1 through 3.1.1.1.7.

3.1.1.1.1 Name Subobject

This subobject contains all the properties relating to the contact’s name. It parses the name and keeps all the properties synchronized with one another.

3.1.1.1.2 E-Mail Subobjects

These subobjects contain e-mail properties. There are six such objects; one for each of the three e-mail addresses, and one for each of the three fax addresses.

3.1.1.1.3 Physical Address Subobjects

These subobjects contain business address, home address, and other address properties. They parse an address and keep all the properties synchronized with one another. There is also one property that specifies which of the Physical Address subobject is the Mailing Address subobject.

3.1.1.1.4 Event Subobjects

These subobjects contain information about the birthday and anniversary properties and maintain the property that links to the corresponding Appointment objects in the calendar module.

3.1.1.1.5 Professional Subobject

This subobject contains the contact's company, job title, and related properties.

3.1.1.1.6 Business Card Subobject

The Business Card object is a set of related properties that are used to customize the visual representation of a contact. A Business Card object can contain one image and one or more text fields.

3.1.1.1.7 Contact Photo Subobject

The Contact Photo object is a set of related properties that specify an optional image of the person or the entity represented by the Contact object. This image exists independently of the Business Card object, but can be used by the Business Card object.

3.1.1.2 Personal Distribution Lists

Personal Distribution List objects extend the Message object specified in [MS-OXCMSG]. Conceptually, personal distribution lists represent a list of e-mail addresses.

3.1.1.3 Contact Folders

Contact folders are folders identified by a special value that are used to store Contact objects and Personal Distribution List objects.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Contact Object Events

3.1.4.1.1 Creating a Contact

To create a Contact object, the server or client creates a Message object as specified in [MS-OXCMSG], sets properties in accordance with the requirements in section 2.2.1 and [MS-OXCPRPT], and saves the resulting Message object as specified in [MS-OXCMSG].

3.1.4.1.2 Deleting a Contact

Contact objects have no special semantics in relation to deletion beyond what is specified in [MS-OXCFOLD].

3.1.4.1.3 Modifying a Contact

When modifying a Contact object, the client or server opens a Message object as specified in [MS-OXCMSG], modifies any of the properties in accordance with the requirements in section 2, sections 3.1.4.1.3.1 through 3.1.4.1.3.7, and [MS-OXCPRPT], and saves the Message object as specified in [MS-OXCMSG].

3.1.4.1.3.1 Modifying a Contact Name Property

Contact name properties MUST be modified at the same time as to preserve their relationship. The following table specifies the properties that SHOULD be updated if another property is changed (according to the definition of each property in section 2.2.1 of this document). More than one condition can be met at the same time.

Changed property	Additional conditions	Properties that SHOULD <28> be updated
PidTagGeneration	None	PidTagDisplayName, PidTagNormalizedSubject
PidTagGeneration or PidTagSurname or PidTagMiddleName or PidTagGivenName or PidTagDisplayName or PidTagCompanyName	Value of the PidLidFileUnderId property is not 0x00000000 or 0xffffffff	PidLidFileUnder
PidTagDisplayNamePrefi x	None	PidTagDisplayName
PidTagSurname or PidTagMiddleName or	None	PidTagDisplayName, PidTagNormalizedSubject,

Changed property	Additional conditions	Properties that SHOULD <28> be updated
PidTagGivenName		PidTagInitials
PidTagDisplayName	None	PidTagDisplayNamePrefix, PidTagGivenName, PidTagMiddleName, PidTagSurname, PidTagNormalizedSubject, PidTagGeneration

The application SHOULD also update **PidLidFax1EmailAddress**, **PidLidFax2EmailAddress**, or **PidLidFax3Address** as needed to reflect any updates to the contact name properties if the contact has a primary fax, business fax, or home fax address, respectively.

3.1.4.1.3.2 Modifying a Physical Address Property

Physical address properties MUST be modified at the same time as to preserve their relationship.

The following table specifies the properties that SHOULD be updated if another property is changed (according to the definition of each property in section 2.2.1 of this document). More than one condition can be met at the same time.

Changed property	Properties that SHOULD <29> be updated
PidLidPostalAddressId	PidTagPostalAddress, PidTagStreetAddress, PidTagLocality, PidTagStateOrProvince, PidTagPostalCode, PidTagCountry, PidLidAddressCountryCode
PidLidWorkAddress	PidLidWorkAddressStreet, PidLidWorkAddressCity, PidLidWorkAddressState, PidLidWorkAddressPostalCode, PidLidWorkAddressCountry, PidLidWorkAddressCountryCode
PidLidHomeAddress	PidTagHomeAddressStreet, PidTagHomeAddressCity, PidTagHomeAddressStateOrProvince, PidTagHomeAddressPostalCode, PidTagHomeAddressCountry, PidLidHomeAddressCountryCode

Changed property	Properties that SHOULD <29> be updated
PidLidOtherAddress	PidTagOtherAddressStreet, PidTagOtherAddressCity, PidTagOtherAddressStateOrProvince, PidTagOtherAddressPostalCode, PidTagOtherAddressCountry, PidLidOtherAddressCountryCode
PidTagHomeAddressStreet	PidLidHomeAddress
PidLidWorkAddressStreet	PidLidWorkAddress
PidTagOtherAddressStreet	PidLidOtherAddress
PidTagHomeAddressCity	PidLidHomeAddress
PidLidWorkAddressCity	PidLidWorkAddress
PidTagOtherAddressCity	PidLidOtherAddress
PidTagHomeAddressStateOrProvince	PidLidHomeAddress
PidLidWorkAddressState	PidLidWorkAddress
PidTagOtherAddressStateOrProvince	PidLidOtherAddress
PidTagHomeAddressPostalCode	PidLidHomeAddress
PidLidWorkAddressPostalCode	PidLidWorkAddress
PidTagOtherAddressPostalCode	PidLidOtherAddress
PidTagHomeAddressCountry	PidLidHomeAddress
PidLidWorkAddressCountry	PidLidWorkAddress
PidTagOtherAddressCountry	PidLidOtherAddress
PidLidHomeAddressCountryCode	PidLidHomeAddress
PidLidWorkAddressCountryCode	PidLidWorkAddress

Changed property	Properties that SHOULD <29> be updated
PidLidOtherAddressCountryCode	PidLidOtherAddress

PidTagPostalAddress, PidTagStreetAddress, PidTagLocality, PidTagStateOrProvince, PidTagPostalCode, PidTagCountry, PidLidAddressCountryCode SHOULD NOT be set without also setting one of the other addresses (either home address, business address, or other address) and setting **PidLidPostalAddress**.

If the home address, business address, or other address is updated, then **PidLidPostalAddress** is set and the physical address that it maps to is changed, then the Mailing Address MUST be updated if appropriate. See section 2.2.1.3.9 for more details.

3.1.4.1.3.3 Modifying an Electronic Address Property

Electronic address properties MUST be modified at the same time as to preserve their relationship. The following table specifies the properties that SHOULD be updated if another property is changed (according to the definition of each property in section 2.2.1 of this document). More than one condition can be met at the same time. The following table includes property names for the primary e-mail address only (**PidLidEmail1**), but the same logic also applies to **PidLidEmail2** and **PidLidEmail3** properties.

Changed property	Additional conditions	Properties that SHOULD <30> be updated
PidLidEmail1OriginalEntryID	PidLidEmail1OriginalEntryID is now either an empty PtypBinary , or it is not set.	PidLidEmail1DisplayName, PidLidEmail1EmailAddress, PidLidEmail1AddressType
	PidLidEmail1OriginalEntryID is now a non-empty PtypBinary .	PidLidEmail1EmailAddress, PidLidEmail1AddressType, PidLidEmail1DisplayName, PidLidEmail1OriginalDisplayName
PidLidEmail1DisplayName	PidLidEmail1OriginalEntryID is a one-off EntryId	PidLidEmail1OriginalEntryID
PidLidEmail1AddressType	PidLidEmail1OriginalEntryID is a one-off EntryId	PidLidEmail1OriginalEntryID

Changed property	Additional conditions	Properties that SHOULD <30> be updated
PidLidEmail1EmailAddress	PidLidEmail1OriginalEntryID is a one-off EntryId	PidLidEmail1OriginalEntryID
	None	PidLidEmail1OriginalDisplayName
PidLidEmail1OriginalDisplayName	None	None

PidLidEmail1OriginalDisplayname SHOULD only be updated when the other e-mail properties are updated.

3.1.4.1.3.4 Updating a FAX number

The electronic fax properties MUST be modified at the same time to preserve their relationship. When one of the fax numbers (**PidTagPrimaryFaxNumber**, **PidTagBusinessFaxNumber**, **PidTagHomeFaxNumber**) is defined, the following properties MUST be set, as specified in section 2.2.1 and the following table.

The fax properties SHOULD only be updated when the corresponding fax number is updated.

Fax number	Properties that SHOULD <31> be updated or defined
PidTagPrimaryFaxNumber	PidLidFax1AddressType , PidLidFax1EmailAddress , PidLidFax1OriginalDisplayName , PidLidFax1OriginalEntryID
PidTagBusinessFaxNumber	PidLidFax2AddressType , PidLidFax2EmailAddress , PidLidFax2OriginalDisplayName , PidLidFax2OriginalEntryID
PidTagHomeFaxNumber	PidLidFax3AddressType , PidLidFax3EmailAddress , PidLidFax3OriginalDisplayName , PidLidFax3OriginalEntryID

3.1.4.1.3.5 Modifying an Event Property

Event properties MUST be modified at the same time to preserve their relationship.

The following table specifies the properties that SHOULD be updated or defined if another property is changed.

Changed property	Conditions	Properties that SHOULD<32> be updated or defined
PidTagBirthday	None	PidLidReferenceEntryId , PidLidBirthdayEventEntryId See the following paragraph for more details.
PidTagWeddingAnniversary	None	PidLidReferenceEntryId , PidLidAnniversaryEventEntryId See the following paragraph for more details.
PidLidBirthdayEventEntryId	None	PidTagBirthday
PidLidAnniversaryEventEntryId	None	PidTagWeddingAnniversary
PidTagNormalizedSubject	PidLidBirthdayEventEntryId is set or PidLidAnniversaryEventEntryId is set	Update the value of the PidTagSubjectPrefix property on the corresponding Appointment object to match the name of the contact.
PidLidPrivate	PidLidBirthdayEventEntryId is set or PidLidAnniversaryEventEntryId is set	Update the value of the PidLidPrivate property on the corresponding Appointment object to match the one on the contact.

When the property **PidTagBirthday** is updated, the client SHOULD update the appointment associated with this contact's birthday (**PidLidBirthdayEventEntryId**) to match the time specified in **PidTagBirthday**. If no appointment has been created, the client SHOULD create an Appointment object, save the Appointment object's EntryId to **PidLidBirthdayEventEntryId**, and link the Appointment object to the Contact object using

PidLidContactLinkEntry, **PidLidContactSearchKey**, **PidLidContactLinkName**, as specified in [MS-OXCMSG].

When the property **PidTagWeddingAnniversary** is updated, the client SHOULD update the appointment associated with this contact's anniversary (**PidLidAnniversaryEventEntryId**) to match the time specified in **PidTagWeddingAnniversary**. If no appointment has been created, the client SHOULD create an Appointment object, save the Appointment object's EntryId to **PidLidAnniversaryEventEntryId**, and link the Appointment object to the Contact object using **PidLidContactLinkEntry**, **PidLidContactSearchKey**, **PidLidContactLinkName**, as specified in [MS-OXCMSG].

3.1.4.1.3.6 Modifying a Business Card Property

Business card properties MUST be modified at the same time as to preserve their relationship. The following table specifies which properties MUST be updated when another property is updated.

Changed property	Conditions	Properties that MUST be updated
PidLidBusinessCardCardPicture	On removal or addition	PidLidBusinessCardDisplayDefinition's ImageSource MUST be updated to account for the card picture, as specified in section 2.2.2.7.2.1.
PidLidBusinessCardDisplayDefinition	If the ImageSource section of the buffer changes	PidLidBusinessCardCardPicture MUST be added or removed

3.1.4.1.3.7 Modifying a Contact Photo Property

Changed property	Conditions	Properties that MUST be updated
PidLidBusinessCardCardPicture	On removal or addition	PidLidBusinessCardDisplayDefinition's ImageSource MUST be updated to account for the Card Picture, as specified in section 2.2.2.7.2.1.
PidLidBusinessCardDisplayDefinition	If the ImageSource section of the buffer changes	PidLidBusinessCardCardPicture MUST be added or removed

3.1.4.2 Personal Distribution List Events

The following events pertain to Personal Distribution List objects.

3.1.4.2.1 Creating a Personal Distribution List

To create a Contact object, the server or client creates a Message object as specified in [MS-OXCMSG], sets properties in accordance with the requirements in section 2.2.2 and [MS-OXCPRPT], and saves the resulting Message object as specified in [MS-OXCMSG].

3.1.4.2.2 Deleting a Personal Distribution List

Personal Distribution List objects have no special semantics in relation to deletion beyond what is specified in [MS-OXCFOLD].

3.1.4.2.3 Modifying a Personal Distribution List

When modifying a Personal Distribution List object, the client or server opens a Message object as specified in [MS-OXCMSG], modifies any of the properties in accordance with the requirements in section 2, sections 3.1.4.2.3.1 through 3.1.4.2.3.4, and [MS-OXCPRPT], and saves the Message object as specified in [MS-OXCMSG].

It is possible that personal distribution lists contain circular references to one another. Such circular references SHOULD be avoided but are permitted under this protocol, and the application MUST be able to handle circular references.

3.1.4.2.3.1 Naming a Personal Distribution List

When the application changes the name of a personal distribution list, either by user input or by other means, the following properties MUST be updated.

Property	Value
PidTagDisplayName	The name of the personal distribution list.
PidLidDistributionListName	The name of the personal distribution list.
PidLidFileUnder	The name of the personal distribution list.
PidLidFileUnderId	This MUST be set to 0xffffffff.
PidTagNormalizedSubject	This MUST be set to the same value as PidLidDistributionListName .

3.1.4.2.3.2 Adding a Member to a Personal Distribution List

Personal Distribution Lists SHOULD NOT contain duplicate entries for the same members. Whenever a member is added to a personal distribution list, the application SHOULD check

that the new member isn't already in the list. Then the application **MUST** update **PidLidDistributionListMembers** and **PidLidDistributionListOneOffMembers**.

The **EntryId** corresponding to the member **MUST** be added to **PidLidDistributionListMembers**, and the one-off **EntryId** corresponding to the member **MUST** be added to **PidLidDistributionListOneOffMembers**. The new values **MUST** be added such that the entry is in the same position of both multi-valued properties.

After updating these properties, the application **SHOULD** update the **PidLidDistributionListChecksum** property, as specified in section 3.1.4.2.3.4.

3.1.4.2.3.3 Removing a Member from a Personal Distribution List

Whenever a member is removed from a personal distribution list, the application **MUST** update **PidLidDistributionListMembers** and **PidLidDistributionListOneOffMembers**.

The **EntryId** corresponding to the member **MUST** be removed from **PidLidDistributionListMembers**, and the one-off **EntryId** corresponding to the member **MUST** be removed from **PidLidDistributionListOneOffMembers**.

After updating these properties, the application **SHOULD** update the **PidLidDistributionListChecksum** property, as specified in section 3.1.4.2.3.4.

3.1.4.2.3.4 Updating the Checksum of a Personal Distribution List

Whenever a member is added to or removed from a personal distribution list, the value of the property **PidLidDistributionListChecksum** **SHOULD** <33> be updated. The checksum **MUST** be calculated starting with a seed value of 0x00000000 and then iterating through each byte of the each **PtypBinary** value in the multi-valued **PidLidDistributionListMembers** property. Use the following algorithm to update **PidLidDistributionListChecksum** value (where CRC32 is defined in [ISO/IEC 8802-3:2000]).

```
SET CheckSum = 0
FOR EACH PtypBinaryValue in PidLidDistributionListMembers
    FOR EACH byteValue in PtypBinaryValue
        Checksum = CRC32(Checksum, byteValue)
    ENDFOR
ENDFOR
```

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Creating a Contact

A user creates a contact with a name (Jacqueline Haddad), e-mail address (someone@example.com), and phone number (555-1234), then saves it. The following is a description of what a client might do to accomplish the user's intentions and the responses a server might return.

Before manipulating Contact objects, the client needs to ask the server to perform a mapping from named properties to property identifiers, using **RopGetPropertyIdsFromNames**:

Property	Property set GUID	Name ID
PidLidFileUnderList	{00062004-0000-0000-c000-000000000046}	0x8026
PidLidAutoLog	{00062004-0000-0000-c000-000000000046}	0x8025
PidLidAddressBookProviderEmailList	{00062004-0000-0000-c000-000000000046}	0x8028
PidLidAddressBookProviderArrayType	{00062004-0000-0000-c000-000000000046}	0x8029
PidLidFileUnder	{00062004-0000-0000-c000-000000000046}	0x8005
PidLidFileUnderId	{00062004-0000-0000-c000-000000000046}	0x8006
PidLidContactCharSet	{00062004-0000-0000-c000-000000000046}	0x8023
PidLidEmail1DisplayName	{00062004-0000-0000-c000-000000000046}	0x8080
PidLidEmail1AddressType	{00062004-0000-0000-c000-000000000046}	0x8082
PidLidEmail1EmailAddress	{00062004-0000-0000-c000-000000000046}	0x8083
PidLidEmail1OriginalDisplayName	{00062004-0000-0000-c000-000000000046}	0x8084
PidLidEmail1OriginalEntryID	{00062004-0000-0000-c000-000000000046}	0x8085

The server returns the following property IDs in response to **RopGetPropertyIdsFromNames**:

Property	Property ID
PidLidFileUnderList	0x827b
PidLidAutoLog	0x8280
PidLidAddressBookProviderEmailList	0x81d4
PidLidAddressBookProviderArrayType	0x81d5
PidLidFileUnder	0x8016
PidLidFileUnderId	0x81da
PidLidContactCharSet	0x8286
PidLidEmail1DisplayName	0x 8013
PidLidEmail1AddressType	0x81ce
PidLidEmail1EmailAddress	0x 8010
PidLidEmail1OriginalDisplayName	0x801f
PidLidEmail1OriginalEntryID	0x81cf

To create a Contact object, the client uses **RopCreateMessage**. The server returns a success code and a handle to a Message object.

After the user has input his content for the Contact object, the client uses **RopSetProperties** to transmit the user's data to the server.

Property	Property type	Data	Meaning
PidTagDisplaynamePrefix	PtypString	00 00	""
PidTagSurname	PtypString	48 00 61 00 64 00 64 00 61 00 64 00 00 00	"Haddad"
PidTagMiddleName	PtypString	00 00	""
PidTagGivenName	PtypString	4a 00 61 00 63 00 71 00 75 00 65 00 6c 00 69 00 6e 00 65 00 00 00	"Jacqueline"
PidLidFileUnderList	PtypMultipleInteger32	05 00 00 00 17 80 00 00 37 80 00 00 16 3a 00 00 19 80 00 00 18 80 00 00	cValues: 0x00000005 lpl: {0x00008017; 0x00008037; 0x00003a16; 0x00008019; 0x00008018}
PidLidAddressBookProviderEmailList	PtypMultipleInteger32	01 00 00 00 00 00 00 00	cValues: 0x00000001 lpl: {0x00000000}

Property	Property type	Data	Meaning
PidLidAddressBookProviderArrayType	PtypInteger32	0x00000001	Email1 is defined
PidLidFileUnder	PtypString	48 00 61 00 64 00 64 00 61 00 64 00 2c 00 20 00 4a 00 61 00 63 00 71 00 75 00 65 00 6c 00 69 00 6e 00 65 00 00 00	"Haddad, Jacqueline"
PidLidFileUnderId	PtypInteger32	0x00008017	PidLidFileUnder SHOULD be "<PidTagSurname>,<PidTagGivenName>,<PidTagMiddleName>"
PidTagGeneration	PtypString	00 00	""
PidTagInitials	PtypString	4a 00 2e 00 48 00 2e 00 00 00	"J.H."
PidLidAutoLog	PtypBoolean	0x00	FALSE
PidTagBusinessTelephoneNumber	PtypString	35 00 35 00 35 00 2d 00 31 00 32 00 33 00 34 00 00	"555-1234"
PidLidContactCharSet	PtypInteger32	0x00000100	US Character set
PidTagDisplayName	PtypString	4a 00 61 00 63 00 71 00 75 00 65 00 6c 00 69 00 6e 00 65 00 20 00 48 00 61 00 64 00 64 00 61 00 64 00 00 00	"Jacqueline Haddad"
PidLidEmailDisplayName	PtypString	75 00 73 00 65 00 72 00 31 00 30 00 20 00 28 00 75 00 73 00 65 00 72 00 31 00 30 00 40 00 73 00 7a 00 66 00 6b 00 75 00 6b 00 2d 00 64 00 6f 00 6d 00 2e 00 65 00 78 00 61 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 29 00 00 00	"user10 (user10@szfkuk-dom.example.com)"
PidLidEmailAddressType	PtypString	45 00 58 00 00 00	"EX"
PidLidEmailEmailAddress	PtypString	2f 00 6f 00 3d 00 46 00 69 00 72 00 73 00 74 00 20 00 4f 00 72 00 67 00 61 00 6e 00 69 00 7a 00 61 00 74 00 69 00 6f 00 6e 00 2f 00 6f 00 75 00 3d 00 45 00 78 00 63 00 68 00 61 00 6e 00 67 00 65 00 20 00 41 00 64 00 6d 00 69 00 6e 00 69 00	"/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=user10"

Property	Property type	Data	Meaning
		73 00 74 00 72 00 61 00 74 00 69 00 76 00 65 00 20 00 47 00 72 00 6f 00 75 00 70 00 20 00 28 00 46 00 59 00 44 00 49 00 42 00 4f 00 48 00 46 00 32 00 33 00 53 00 50 00 44 00 4c 00 54 00 29 00 2f 00 63 00 6e 00 3d 00 52 00 65 00 63 00 69 00 70 00 69 00 65 00 6e 00 74 00 73 00 2f 00 63 00 6e 00 3d 00 75 00 73 00 65 00 72 00 31 00 30 00 00 00	
PidLidEmailOriginalDisplayName	PtypString	75 00 73 00 65 00 72 00 31 00 30 00 40 00 73 00 7a 00 66 00 6b 00 75 00 6b 00 2d 00 64 00 6f 00 6d 00 2e 00 65 00 78 00 61 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 00 00	"user10@szfkuk-dom.example.com"
PidLidEmailOriginalEntryID	PtypBinary	7d 00 00 00 DC A7 40 C8 C0 42 10 1A B4 B9 08 00 2B 2F E1 82 01 00 00 00 00 00 00 00 2F 6F 3D 46 69 72 73 74 20 4F 72 67 61 6E 69 7A 61 74 69 6F 6E 2F 6F 75 3D 45 78 63 68 61 6E 67 65 20 41 64 6D 69 6E 69 73 74 72 61 74 69 76 65 20 47 72 6F 75 70 20 28 46 59 44 49 42 4F 48 46 32 33 53 50 44 4C 54 29 2F 63 6E 3D 52 65 63 69 70 69 65 6E 74 73 2F 63 6E 3D 75 73 65 72 31 30 00	Size: 125 bytes@..B+//o=Fir st Organ ization/ ou=Excha nge Admi nistrati ve Group (FYDIBO HF23SPDL T)/cn=Re cipients /cn=user 10.
PidTagRtfCompressed	PtypBinary	e6 14 E2 14 00 00 93 52 00 00 4C 5A 46 75 62 F8 7E BB 07 00 06 01 01 0B 60 6E 67 31 30 32 66 35 00 64 00 72 63 70 0D D0 0E 00 32 05 0C 60 63 0D 44 66 33 31 35 30 42 37 00 F5 73 74 73 68 05 70 62 74 ... (remainder of property excluded for brevity)	Size: 5350 bytesR.. LZFub.~.`n g102f5.d .rcp.... 2..`c.Df 3150B7.. stsh.pbt
PidTagMessageClass	PtypString	49 00 50 00 4d 00 2e 00 43 00 6f 00 6e 00 74 00 61 00 63 00 74 00 00 00	"IPM.Contact"
PidTagIconIndex	PtypInteger32	00 20 00 00	512
PidTagSubjectPrefix	PtypString	00	""

Property	Property type	Data	Meaning
PidTagNormalizedSubject	PtypString	4a 00 61 00 63 00 71 00 75 00 65 00 6c 00 69 00 6e 00 65 00 20 00 48 00 61 00 64 00 64 00 61 00 64 00 00 00	"Jacqueline Haddad"

When the user is ready to save their changes, the client uses **RopSaveChanges** to commit the properties on the server.

The values of some properties will change during the execution of **RopSaveChanges**; however, none of the properties that change are specified contact properties.

4.2 Creating a Personal Distribution List

A user creates a personal distribution list that contains the previous sample contact and another e-mail address (user1@example.com). The following is a description of what a client might do to accomplish the user's intentions and the responses a server might return.

Before manipulating Contact objects, the client needs to ask the server to perform a mapping from named properties to property identifiers, using **RopGetPropertyIdsFromNames**:

Property	Property set GUID	Name ID
PidLidAutoLog	{00062004-0000-0000-c000-000000000046}	0x8025
PidLidAddressBookProviderArrayType	{00062004-0000-0000-c000-000000000046}	0x8029
PidLidFileUnder	{00062004-0000-0000-c000-000000000046}	0x8005
PidLidFileUnderId	{00062004-0000-0000-c000-000000000046}	0x8006
PidLidDistributionListName	{00062004-0000-0000-c000-000000000046}	0x8053
PidLidDistributionListChecksum	{00062004-0000-0000-c000-000000000046}	0x804c
PidLidDistributionListMembers	{00062004-0000-0000-c000-000000000046}	0x8055
PidLidDistributionListOneOffMembers	{00062004-0000-0000-c000-000000000046}	0x8054

The server returns the following property ids in response to **RopGetPropertyIdsFromNames**:

Property	Property ID
PidLidAutoLog	0x8280
PidLidAddressBookProviderArrayType	0x81d5
PidLidFileUnder	0x8016
PidLidFileUnderId	0x81da
PidLidDistributionListName	0x81c9
PidLidDistributionListChecksum	0x81c7
PidLidDistributionListMembers	0x81c8
PidLidDistributionListOneOffMembers	0x81ca

To create a Personal Distribution List object, the client uses RopCreateMessage. The server returns a success code and a handle to a Message object.

After the user has input content for the Personal Distribution List object, the client uses RopSetProperties to transmit his data to the server.

Property	Property type	Data	Meaning
PidTagDisplayNamePrefix	PtypString	00 00	""
PidLidAddressBookProviderArrayType	PtypInteger32	0x00000000	0
PidLidFileUnder	PtypString	46 00 72 00 69 00 65 00 6e 00 64 00 73 00 00 00	"Friends"
PidLidFileUnderId	PtypInteger32	0xffffffff	-1
PidLidAutoLog	PtypBoolean	0x00	FALSE
PidTagDisplayName	PtypString	46 00 72 00 69 00 65 00 6e 00 64 00 73 00 00 00	Friends
PidLidDistributionListName	PtypString	46 00 72 00 69 00 65 00 6e 00 64 00 73 00 00 00	Friends
PidLidDistributionListChecksum	PtypInteger32	0xd4b0223a	-726654406
PidLidDistributionListMembers	PtypMultipleBinary	02 00 00 00 64 00 00 00 00 00 81 2b 1f a4 be a3 10 19 9d 6e 00 dd 01 0f 54 02 00 00 01 80 54 00 68 00 6f 00 6d 00 61 00 73 00 20 00 48 00 61 00 6d 00 62 00 6f 00 72 00 67 00 00 00 53 00 4d 00 54 00 50 00 00 00 75 00 73 00 65 00 72 00 31 00 40 00 65 00 78 00 61 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 00 00 5b 00 00 00 00 00 c0 91 ad d3 51 9d cf 11 a4 a9 00 aa 00 47 fa a4 c3 00 00 00 00 c2 42 42 82 60 00 16 47 ad a9 5c 07 77 b7 4b e2 07 00 61 2a 7b ab 49 f6 4e 4b 9c 52 db fb 5a 53 aa 1c 00 00 00 f0 4e a6 00 00 61 2a 7b ab 49 f6 4e 4b 9c 52 db fb 5a 53 aa 1c 00 00 00 f0 d5 b0 00 00	2 BLOBs to follow BLOB1 Size: 100 bytes+..n.. ..T..... T.h.o.m. a.s. .H. a.m.b.o. r.g...S. M.T.P... u.s.e.r. l.@.e.x. a.m.p.l. e...C.o. m... BLOB2 Size:91 bytes Q..... .G..... ..BB.`.. G..\w.K ...a*{.I .NK.R..Z S.....N ...a*{.I .NK.R..Z S..... ...
PidLidDistributionListOnlineMembers	PtypMultipleBinary	02 00 00 00 64 00 00 00 00 00 81 2b 1f a4 be a3 10 19 9d 6e 00 dd 01 0f 54 02 00 00 01 80	2 BLOBs to follow BLOB1 Size: 100 bytes+..n.. ..T.....

Property	Property type	Data	Meaning
		54 00 68 00 6f 00 6d 00 61 00 73 00 20 00 48 00 61 00 6d 00 62 00 6f 00 72 00 67 00 00 00 53 00 4d 00 54 00 50 00 00 00 75 00 73 00 65 00 72 00 31 00 40 00 65 00 78 00 61 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 00 00 ac 00 00 00 00 00 81 2b 1f a4 be a3 10 19 9d 6e 00 dd 01 0f 54 02 00 00 01 80 75 00 73 00 65 00 72 00 31 00 30 00 20 00 28 00 75 00 73 00 65 00 72 00 31 00 30 00 40 00 73 00 7a 00 66 00 6b 00 75 00 6b 00 2d 00 64 00 6f 00 6d 00 2e 00 65 00 78 00 6a 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 29 00 00 00 53 00 4d 00 54 00 50 00 00 00 75 00 73 00 65 00 72 00 31 00 30 00 40 00 73 00 7a 00 66 00 6b 00 75 00 6b 00 2d 00 64 00 6f 00 6d 00 2e 00 65 00 78 00 6a 00 6d 00 70 00 6c 00 65 00 2e 00 63 00 6f 00 6d 00 00 00	T.h.o.m. a.s. .H. a.m.b.o. r.g...S. M.T.P... u.s.e.r. l.@.e.x. a.m.p.l. e...C.o. m... BLOB2 Size: 172 bytes+..n.. ..T..... u.s.e.r. l.0. .(. u.s.e.r. l.0.@.s. z.f.k.u. k.-.d.o. m...e.x. a.m.p.l. e...c.o. m.)...S. M.T.P... u.s.e.r. l.0.@.s. z.f.k.u. k.-.d.o. m...e.x. a.m.p.l. e...C.o. m...
PidTagMessageClass	PtypString	49 00 50 00 4d 00 2e 00 44 00 69 00 73 00 74 00 4c 00 69 00 73 00 74 00 00 00	"IPM.DistList"
PidTagIconIndex	PtypInteger32	0x00000202	514
PidTagSubjectPrefix	PtypString	00 00	""
PidTagNormalizedSubject	PtypString	46 00 72 00 69 00 65 00 6e 00 64 00 73 00 00 00	"Friends"

When the user is ready to save their changes, the client uses **RopSaveChanges** to commit the properties on the server.

The values of some properties will change during the execution of **RopSaveChanges**; however, none of the properties that change are specified contact properties.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the Contact Object Protocol Specification. General security considerations pertaining to the underlying transport apply, as specified in [MS-OXCMSG] and [MS-OXCPRPT].

5.2 Index of Security Parameters

None.

6 Appendix A: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Office 2003 with Service Pack 3 applied
- Exchange 2003 with Service Pack 2 applied
- Office 2007 with Service Pack 1 applied
- Exchange 2007 with Service Pack 1 applied

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Office/Exchange does not follow the prescription.

<1> Section 2.2: Outlook 2003 SP3 and Outlook 2007 SP1 set the following properties regardless of user input; their values have no meaning in the context of this protocol.

PidLidAgingDontAgeMe, PidLidCurrentVersion, PidLidCurrentVersionName, PidLidPrivate, PidLidValidFlagStringProof, PidTagAlternateRecipientAllowed, PidTagclientSubmitTime, PidTagDeleteAfterSubmit, PidTagImportance, PidTagMessageDeliveryTime, PidTagMessageLocaleID, PidTagOriginatorDeliveryReportRequested, PidLidSideEffects, PidTagPriority, PidTagReadReceiptRequested, PidTagSensitivity

<2> Section 2.2: Outlook 2007 SP1 sets the following properties regardless of user input; their values have no meaning in the context of this protocol: **PidLidReminderSet,**

PidLidTaskActualEffort, PidLidTaskComplete, PidLidTaskAssigner, PidLidTaskAcceptanceState, PidLidTaskEstimatedEffort, PidLidTaskFFixOffline, PidLidTaskFRecurring, PidLidTaskMode, PidLidTaskNoCompute, PidLidTaskOrdinal, PidLidTaskOwnership, PidLidTaskRole, PidLidTaskState, PidLidTaskStatus, PidLidTaskVersion, and PidLidTeamTask

<3> Section 2.2: Outlook 2003 SP3 and Outlook 2007 SP1 sometimes set the following properties; they SHOULD be ignored by the client and the server:

PidLidEmail1RichTextFormat, **PidLidEmail2RichTextFormat**,
PidLidEmail3RichTextFormat, and **PidLidReferredBy**.

<4> Section 2.2.1.1.12: This table shows the format that the **PtypString** property **PidLidFileUnder** SHOULD be in, if all of the properties are present. If a property is not present, the separator characters surrounding it MAY be removed by the application.

In cases where the format string shows **PidTagSurname** before **PidTagGivenName**, Outlook 2003 SP3 and Outlook 2007 SP1 move portions of **PidTagSurname** such as “van ” or “de ” to right after **PidTagMiddleName** if that property is present and part of the format string, or to right after **PidTagGivenName**, if that property is present and part of the format string.

<5> Section 2.2.1.1.13: The **PidLidFileUnderList** property is set by Outlook 2003 SP3 and Outlook 2007 SP1 but never used by either Outlook or Exchange

<6> Section 2.2.1.2: Outlook 2003 SP3 and Outlook 2007 SP1 set the **PidLidAddressBookProviderEmailList** and **PidLidAddressBookProviderArrayType** and Primary Fax, Business Fax, or Home Fax properties only if they are able to interpret the value of the **PidTagPrimaryFaxNumber**, **PidTagBusinessFaxNumber**, or **PidTagHomeFaxNumber** property, respectively, as an actual fax phone number.

Exchange 2007 SP1 sets the **PidLidAddressBookProviderEmailList** and **PidLidAddressBookProviderArrayType** properties whenever the **FaxNumber** string property is set.

Exchange 2003 SP2 always sets the **PidLidAddressBookProviderArrayType** property to 0x00000000 and never sets the **PidLidAddressBookProviderEmailList** property unless an e-mail address is defined for the contact.

<7> Section 2.2.1.2.2: Common address types used by Outlook and Exchange are “SMTP” and “EX”, but can be any of the returned from **PropAddressTypes** or even a third-party defined string.

<8> Section 2.2.1.2.3: Each address type has a specific format. The format of “EX” e-mail addresses is specified in [RFC2247]. The format of “SMTP” e-mail addresses is specified in [RFC821]. The e-mail type string is intended to provide an avenue for various third-party delivery mechanisms to define their own e-mail format. A Contact object is simply used as storage for these values and e-mail types and e-mail addresses have no special meaning in the context of this protocol.

<9> Section 2.2.1.2.11: Outlook 2003 SP3 and Outlook 2007 SP1 set this multi-valued **LONG** property to contain the three values 0x0008080, 0x0008090, and 0x00080A0 whenever one e-mail address is defined. Exchange does not set this property.

-
- <10> Section 2.2.1.2.12: The fact that there are two properties (**PidLidAddressBookProviderEmailList** and **PidLidAddressBookProviderArrayType**) containing similar information is an application optimization.
- <11> Section 2.2.1.3.1: Outlook and Exchange also use this property to store the post office box part of the address, if it exists.
- <12> Section 2.2.1.3.7: Outlook and Exchange do not use the post office box properties when generating the full address, or when parsing the full address into components. Instead, they place the post office box data as part of the street address.
- <13> Section 2.2.1.6.4: Outlook 2003 SP3 and Outlook 2007 SP1 always set this property to an empty string.
- <14> Section 2.2.1.7: The business card properties are only implemented in Outlook 2007 SP1 and are ignored by Outlook 2003 SP3 and by the server.
- <15> Section 2.2.1.7.1.1.1: Outlook 2007 SP1 sets the **MajorVersion** to 0x03 and the **MinorVersion** to 0x00.
- <16> Section 2.2.1.8: Contact Photo properties are only used by Outlook 2003 SP3, Outlook 2007 SP1 and Exchange 2007 SP1.
- <17> Section 2.2.1.8.2: If there is more than one attachment that has this property set, the application is free to pick any of the marked attachments as the contact photo.
- <18> Section 2.2.1.9.1: This property can be used to detect when a Message object is a copy of an earlier original (when the value of **PidLidReferenceEID** is not the same as the **EntryID** of the Contact object).
- <19> Section 2.2.1.9.11: The value of this property is not used by either the client or the server.
- <20> Section 2.2.1.9.16: The value of this property is not used by either the client or the server.
- <21> Section 2.2.1.9.18: For US English, Outlook sets this value to the more generic 0x00000100, denoting a “western” character set. Exchange does not set or use this property.
- <22> Section 2.2.1.10.1: Outlook uses the value of the **PidTagNormalizedSubject** property as the caption of the window displaying information about this contact. The value of the very similar **PidTagDisplayName** property might contain characters that cannot be displayed as a window caption.
- <23> Section 2.2.2.1.2: For a newly created distribution list without a name at all, Outlook 2003 SP3 and Outlook 2007 SP1 do not set the property **PidLidDistributionListName** to any value. In the same scenario, Outlook 2003 SP3 and Outlook 2007 SP1 set **PidTagDisplayName** to the string containing just a NULL terminator. In every other case, Outlook keeps these two properties the same.

Exchange never sets this property.

<24> Section 2.2.2.1.4: Exchange 2007 SP1 sets this property to 0 when it creates a distribution list.

<25> Section 2.2.2.2.2: Exchange 2003 does not set or update this property

<26> Section 2.2.2.2.3: Exchange 2003 and Exchange 2007 does not set or update this property

<27> Section 2.2.2.3.1: Exchange does not set this property

<28> Section 3.1.4.1.3.1: Not all properties will need to be updated every time. Applications MUST use the specifications in section 2.2.1.1 to determine if an update is required.

<29> Section 3.1.4.1.3.2: Not all properties will need to be updated every time. Applications MUST use the specifications in section 2.2.1.3 to determine if an update is required.

<30> Section 3.1.4.1.3.3: Not all properties will need to be updated every time. Applications MUST use the specifications in section 2.2.1.2 to determine if an update is required.

<31> Section 3.1.4.1.3.4: Not all properties will need to be updated every time. Applications MUST use the specifications in section 2.2.1.2 to determine if an update is required.

<32> Section 3.1.4.1.3.5: Not all properties will need to be updated every time. Applications MUST use the specifications in section 2.2.1.5 to determine if an update is required.

<33> Section 3.1.4.2.3.4: Exchange does not update this property

7 Index

Applicability statement, 7

Glossary, 4

Index of security parameters, 53

Introduction, 4

Messages, 7

 Message syntax, 7

 Transport, 7

Normative references, 5

Office/Exchange behavior, 53

Prerequisites/preconditions, 7

Protocol details, 35

 Client and server details, 35

Protocol examples, 46

 Creating a contact, 46

 Creating a personal distribution list, 50

Protocol overview (synopsis), 6

References, 5

 Informative references, 6

 Normative references, 5

Relationship to other protocols, 7

Security, 53

 Index of security parameters, 53

 Security considerations for implementers, 53

Security considerations for implementers, 53

Standards assignments, 7

Transport, 7

Vendor-extensible fields, 7

Versioning and capability negotiation, 7