

[MS-OXOCNTC]: Contact Object Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1	Major	Initial Availability.
04/25/2008	0.2	Minor	Revised and updated property names and other technical content.
06/27/2008	1.0	Major	Initial Release.
08/06/2008	1.0.1	Editorial	Revised and edited technical content.
09/03/2008	1.0.2	Editorial	Revised and edited technical content.
12/03/2008	1.0.3	Editorial	Revised and edited technical content.
03/04/2009	1.0.4	Editorial	Revised and edited technical content.
04/10/2009	2.0	Major	Updated technical content for new product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	5.0.0	Major	Updated and revised the technical content.
05/05/2010	6.0.0	Major	Updated and revised the technical content.
08/04/2010	6.1	Minor	Clarified the meaning of the technical content.

Contents

1 Introduction	8
1.1 Glossary	8
1.2 References	9
1.2.1 Normative References	9
1.2.2 Informative References	10
1.3 Overview	10
1.4 Relationship to Other Protocols	10
1.5 Prerequisites/Preconditions	11
1.6 Applicability Statement	11
1.7 Versioning and Capability Negotiation	11
1.8 Vendor-Extensible Fields	11
1.9 Standards Assignments	11
2 Messages	12
2.1 Transport	12
2.2 Message Syntax	12
2.2.1 Contact Object Properties	12
2.2.1.1 Contact Name Properties	12
2.2.1.1.1 PidTagNickname	12
2.2.1.1.2 PidTagGeneration	12
2.2.1.1.3 PidTagDisplayNamePrefix	12
2.2.1.1.4 PidTagSurname	12
2.2.1.1.5 PidTagMiddleName	13
2.2.1.1.6 PidTagGivenName	13
2.2.1.1.7 PidTagInitials	13
2.2.1.1.8 PidTagDisplayName	13
2.2.1.1.9 PidLidYomiFirstName	13
2.2.1.1.10 PidLidYomiLastName	13
2.2.1.1.11 PidLidFileUnder	13
2.2.1.1.12 PidLidFileUnderId	13
2.2.1.1.13 PidLidFileUnderList	15
2.2.1.2 Electronic Address Properties	15
2.2.1.2.1 PidLidEmail1DisplayName, PidLidEmail2DisplayName, PidLidEmail3DisplayName	16
2.2.1.2.2 PidLidEmail1AddressType, PidLidEmail2AddressType, PidLidEmail3AddressType	16
2.2.1.2.3 PidLidEmail1EmailAddress, PidLidEmail2EmailAddress, PidLidEmail3EmailAddress	16
2.2.1.2.4 PidLidEmail1OriginalDisplayName, PidLidEmail2OriginalDisplayName, PidLidEmail3OriginalDisplayName	17
2.2.1.2.5 PidLidEmail1OriginalEntryId, PidLidEmail2OriginalEntryId, PidLidEmail3OriginalEntryId	17
2.2.1.2.6 PidTagPrimaryFaxNumber, PidTagBusinessFaxNumber, PidTagHomeFaxNumber	17
2.2.1.2.7 PidLidFax1AddressType, PidLidFax2AddressType, PidLidFax3AddressType	17
2.2.1.2.8 PidLidFax1EmailAddress, PidLidFax2EmailAddress, PidLidFax3EmailAddress	17
2.2.1.2.9 PidLidFax1OriginalDisplayName, PidLidFax2OriginalDisplayName, PidLidFax3OriginalDisplayName	17

2.2.1.2.10	PidLidFax1OriginalEntryId, PidLidFax2OriginalEntryId, PidLidFax3OriginalEntryId.....	17
2.2.1.2.11	PidLidEmailList	18
2.2.1.2.12	PidLidAddressBookProviderEmailList	18
2.2.1.2.13	PidLidAddressBookProviderArrayType.....	18
2.2.1.3	Physical Address Properties.....	19
2.2.1.3.1	PidLidWorkAddressStreet, PidTagHomeAddressStreet, PidTagOtherAddressStreet, PidTagStreetAddress.....	20
2.2.1.3.2	PidLidWorkAddressCity, PidTagHomeAddressCity, PidTagOtherAddressCity, PidTagLocality	20
2.2.1.3.3	PidLidWorkAddressState, PidTagHomeAddressStateOrProvince, PidTagOtherAddressStateOrProvince, PidTagStateOrProvince	20
2.2.1.3.4	PidLidWorkAddressPostalCode, PidTagHomeAddressPostalCode, PidTagOtherAddressPostalCode, PidTagPostalCode	20
2.2.1.3.5	PidLidWorkAddressCountry, PidTagHomeAddressCountry, PidTagOtherAddressCountry, PidTagCountry	20
2.2.1.3.6	PidLidWorkAddressCountryCode, PidLidHomeAddressCountryCode, PidLidOtherAddressCountryCode, PidLidAddressCountryCode	20
2.2.1.3.7	PidLidWorkAddressPostOfficeBox, PidTagHomeAddressPostOfficeBox, PidTagOtherAddressPostOfficeBox, PidTagPostOfficeBox	20
2.2.1.3.8	PidLidWorkAddress, PidLidHomeAddress, PidLidOtherAddress, PidTagPostalAddress	21
2.2.1.3.9	PidLidPostalAddressId	21
2.2.1.4	Telephone Properties.....	21
2.2.1.4.1	PidTagPagerTelephoneNumber	21
2.2.1.4.2	PidTagCallbackTelephoneNumber	22
2.2.1.4.3	PidTagBusinessTelephoneNumber.....	22
2.2.1.4.4	PidTagHomeTelephoneNumber.....	22
2.2.1.4.5	PidTagPrimaryTelephoneNumber	22
2.2.1.4.6	PidTagBusiness2TelephoneNumber	22
2.2.1.4.7	PidTagMobileTelephoneNumber	22
2.2.1.4.8	PidTagRadioTelephoneNumber	22
2.2.1.4.9	PidTagCarTelephoneNumber	22
2.2.1.4.10	PidTagOtherTelephoneNumber	22
2.2.1.4.11	PidTagAssistantTelephoneNumber	22
2.2.1.4.12	PidTagHome2TelephoneNumber	22
2.2.1.4.13	PidTagTtyTddPhoneNumber	22
2.2.1.4.14	PidTagCompanyMainTelephoneNumber	22
2.2.1.4.15	PidTagTelexNumber	23
2.2.1.4.16	PidTagIsdnNumber	23
2.2.1.5	Event Properties	23
2.2.1.5.1	PidTagBirthday	23
2.2.1.5.2	PidLidBirthdayLocal	23
2.2.1.5.3	PidLidBirthdayEventEntryId	23
2.2.1.5.4	PidTagWeddingAnniversary.....	23
2.2.1.5.5	PidLidWeddingAnniversaryLocal	23
2.2.1.5.6	PidLidAnniversaryEventEntryId	23
2.2.1.6	Professional Properties	24
2.2.1.6.1	PidTagTitle	24
2.2.1.6.2	PidTagCompanyName	24
2.2.1.6.3	PidTagDepartmentName	24
2.2.1.6.4	PidLidDepartment.....	24
2.2.1.6.5	PidTagOfficeLocation.....	24

2.2.1.6.6	PidTagManagerName	24
2.2.1.6.7	PidTagAssistant.....	24
2.2.1.6.8	PidLidYomiCompanyName	24
2.2.1.6.9	PidTagProfession	24
2.2.1.7	Business Card Properties	24
2.2.1.7.1	PidLidBusinessCardDisplayDefinition	24
2.2.1.7.1.1	PidLidBusinessCardDisplayDefinition Buffer Format	25
2.2.1.7.1.1.1	MajorVersion and MinorVersion	26
2.2.1.7.1.1.2	TemplateID	26
2.2.1.7.1.1.3	ImageAlignment	27
2.2.1.7.1.1.4	ImageSource	27
2.2.1.7.1.1.5	BackgroundColor.....	27
2.2.1.7.1.1.6	ImageArea	27
2.2.1.7.1.2	FieldInfo Buffer Format	28
2.2.1.7.1.2.1	TextPropertyID	28
2.2.1.7.1.2.2	TextFormat	30
2.2.1.7.1.2.3	LabelFormat	30
2.2.1.7.1.2.4	FontSize	30
2.2.1.7.1.2.5	LabelOffset.....	30
2.2.1.7.1.2.6	ValueFontColor	31
2.2.1.7.1.2.7	LabelFontColor.....	31
2.2.1.7.1.3	ExtraInfo Buffer Format.....	31
2.2.1.7.2	PidLidBusinessCardCardPicture	31
2.2.1.7.3	PidLidContactUserField1, PidLidContactUserField2, PidLidContactUserField3, PidLidContactUserField4	31
2.2.1.8	Contact Photo Properties	31
2.2.1.8.1	PidLidHasPicture	31
2.2.1.8.2	Contact Photo Attachment.....	32
2.2.1.9	Other Contact Properties	32
2.2.1.9.1	PidLidReferenceEntryId	32
2.2.1.9.2	PidTagHobbies	32
2.2.1.9.3	PidTagSpouseName	32
2.2.1.9.4	PidTagLanguage.....	32
2.2.1.9.5	PidTagLocation.....	32
2.2.1.9.6	PidLidInstantMessagingAddress.....	32
2.2.1.9.7	PidTagOrganizationalIdNumber	32
2.2.1.9.8	PidTagCustomerId	33
2.2.1.9.9	PidTagGovernmentIdNumber	33
2.2.1.9.10	PidLidFreeBusyLocation	33
2.2.1.9.11	PidTagAccount	33
2.2.1.9.12	PidLidHtml	33
2.2.1.9.13	PidTagPersonalHomePage.....	33
2.2.1.9.14	PidTagBusinessHomePage	33
2.2.1.9.15	PidTagFtpSite.....	33
2.2.1.9.16	PidTagComputerNetworkName	33
2.2.1.9.17	PidTagChildrensNames	33
2.2.1.9.18	PidLidContactCharacterSet.....	33
2.2.1.9.19	PidLidAutoLog	34
2.2.1.9.20	PidTagGender	34
2.2.1.9.21	PidTagReferredByName	34
2.2.1.9.22	PidLidContactItemData.....	34
2.2.1.9.23	PidTagUserX509Certificate.....	35
2.2.1.9.24	PidLidBilling	35

2.2.1.10	Additional Property Constraints.....	35
2.2.1.10.1	PidTagNormalizedSubject	35
2.2.1.10.2	PidTagMessageClass	35
2.2.2	Personal Distribution List Properties.....	35
2.2.2.1	Personal Distribution List Name Properties	35
2.2.2.1.1	PidTagDisplayName	36
2.2.2.1.2	PidLidDistributionListName	36
2.2.2.1.3	PidLidFileUnder	36
2.2.2.1.4	PidLidFileUnderId	36
2.2.2.2	Personal Distribution List Member Properties	36
2.2.2.2.1	PidLidDistributionListMembers.....	36
2.2.2.2.2	PidLidDistributionListOneOffMembers.....	36
2.2.2.2.3	PidLidDistributionListChecksum	36
2.2.2.2.4	PidLidDistributionListStream	37
2.2.2.2.4.1	PidLidDistributionListStream Buffer Format	37
2.2.2.2.4.2	DistListMemberInfo Format.....	38
2.2.2.2.5	Wrapped EntryId Format.....	40
2.2.2.3	Other Personal Distribution List Properties	41
2.2.2.3.1	PidLidAddressBookProviderArrayType	41
2.2.2.4	Additional Property Constraints	41
2.2.2.4.1	PidTagNormalizedSubject	41
2.2.2.4.2	PidTagMessageClass	41
2.2.3	Contact Folder Syntax.....	42
3	Protocol Details.....	43
3.1	Common Details	43
3.1.1	Abstract Data Model	43
3.1.1.1	Contact Object	43
3.1.1.1.1	Name Subobject	43
3.1.1.1.2	E-Mail Subobjects.....	43
3.1.1.1.3	Physical Address Subobjects	43
3.1.1.1.4	Event Subobjects	43
3.1.1.1.5	Professional Subobject.....	43
3.1.1.1.6	Business Card Subobject.....	43
3.1.1.1.7	Contact Photo Subobject	44
3.1.1.2	Personal Distribution Lists.....	44
3.1.1.3	Contact Folders	44
3.1.2	Timers	44
3.1.3	Initialization	44
3.1.4	Higher-Layer Triggered Events.....	44
3.1.4.1	Contact Object Events	44
3.1.4.1.1	Creating a Contact	44
3.1.4.1.2	Deleting a Contact.....	44
3.1.4.1.3	Modifying a Contact.....	44
3.1.4.1.3.1	Modifying a Contact Name Property	44
3.1.4.1.3.2	Modifying a Physical Address Property	45
3.1.4.1.3.3	Modifying an E-mail Address Property	46
3.1.4.1.3.4	Updating a FAX Number	47
3.1.4.1.3.5	Modifying an Event Property	47
3.1.4.1.3.6	Modifying a Business Card Property	49
3.1.4.1.3.7	Modifying a Contact Photo Property	49
3.1.4.2	Personal Distribution List Events.....	49
3.1.4.2.1	Creating a Personal Distribution List	49

3.1.4.2.2	Deleting a Personal Distribution List	49
3.1.4.2.3	Modifying a Personal Distribution List.....	49
3.1.4.2.3.1	Naming a Personal Distribution List.....	50
3.1.4.2.3.2	Adding a Member to a Personal Distribution List	50
3.1.4.2.3.3	Removing a Member from a Personal Distribution List	50
3.1.4.2.3.4	Updating the Checksum of a Personal Distribution List	50
3.1.5	Message Processing Events and Sequencing Rules.....	51
3.1.6	Timer Events	51
3.1.7	Other Local Events	51
4	Protocol Examples.....	52
4.1	Creating a Contact.....	52
4.2	Creating a Personal Distribution List.....	60
5	Security.....	66
5.1	Security Considerations for Implementers.....	66
5.2	Index of Security Parameters	66
6	Appendix A: Product Behavior	67
7	Change Tracking.....	69
8	Index	71

1 Introduction

This document specifies the Contact Object Protocol, a **Message object** that serves as the basic organizational unit for personal information, such as phone numbers, e-mail addresses, and mailing addresses. It also serves as the basic organizational unit for **collections** of e-mail addresses, which can be used to create mailing lists.

This document extends the Message and Attachment Object Protocol, as specified in [\[MS-OXCMSG\]](#), and specifies the **properties** that are set on **Contact objects** and **Personal Distribution List objects** in addition to the Message object properties specified in [\[MS-OXCMSG\]](#) section 2.2.1. This specification assumes that the reader has familiarity with the concepts and requirements of [\[MS-OXCMSG\]](#). Those concepts and requirements are not repeated in this specification.

1.1 Glossary

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

- .jpg**
- action**
- address book**
- Address Book object**
- address type**
- appointment**
- Appointment object**
- attachment**
- binary large object (BLOB)**
- character set**
- collection**
- condition**
- contact**
- Contact object**
- Coordinated Universal Time (UTC)**
- cyclic redundancy check (CRC) (1)**
- display template**
- distribution list**
- EntryID**
- folder**
- free/busy**
- Global Address List (GAL)**
- handle**
- Journal object**
- little-endian**
- locale**
- mail user**
- mailbox**
- message**
- Message object**
- one-off EntryID**
- Personal Distribution List object**
- portable network graphics (PNG)**
- property (1)**
- property ID**
- remote operation (ROP)**
- Simple Mail Transfer Protocol (SMTP)**

store
subobject
Unicode
Uniform Resource Locator (URL)

The following terms are specific to this document:

BMP: A standard format used to store device-independent and application-independent images. The number of bits per pixel (1, 4, 8, 15, 24, 32, or 64) for a given **BMP** file is specified in a file header. **BMP** files with 24 bits per pixel are common.

enhanced metafile format (EMF): A file format that supports device-independent definitions of images.

EXIF: A file format used for photographs captured by digital cameras. An **EXIF** file contains an image that is compressed according to the **.jpg** specification. An **EXIF** file also contains information about the photograph (date taken, shutter speed, exposure time, and so on) and information about the camera (manufacturer, model, and so on).

GIF: A common format for images that appear on Web pages. **GIFs** work well for line drawings, pictures with blocks of solid color, and pictures with sharp boundaries between colors. **GIFs** are compressed, but no information is lost in the compression process; a decompressed image is exactly the same as the original.

ICO: A file format used for icons in the operating system.

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

TIFF: A flexible and extendable format that is supported by a wide variety of platforms and image-processing applications. **TIFF** files can store images with an arbitrary number of bits per pixel and can employ a variety of compression algorithms.

Windows metafile format (WMF): A file format used by Windows that supports definitions of images.

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

1.2 References

1.2.1 Normative References

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

[ISO/IEC8802-3] 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](#)", April 2008.

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

[MS-OXCICAL] Microsoft Corporation, "[Calendar 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-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, "[Exchange Server Protocols Master Property List](#)", 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., et al., "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.rfc-editor.org/rfc/rfc821.txt>

[RFC959] Postel, J., and Reynolds, J., "FILE TRANSFER PROTOCOL (FTP)", RFC 959, October 1985, <http://www.ietf.org/rfc/rfc959.txt>

1.2.2 Informative References

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

1.3 Overview

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

This protocol is an extension of the Message and Attachment Object Protocol, as specified in [\[MS-OXCMSG\]](#), and specifies the representation of a **contact** or personal **distribution list** in a messaging **store**. This document specifies the properties that are unique to Contact objects and Personal Distribution List objects, and the interdependencies that exist between subsets of these properties. A Contact object can contain subsets of properties that together define a **subobject** stored on a Contact object, 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, as specified in [\[MS-OXCMSG\]](#), which specifies Message objects. [\[MS-OXCMSG\]](#) specifies the **remote operations (ROPs)** used to create and modify Message objects, and the properties that are common to all 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

The Contact Object Protocol assumes that the client has previously acquired a **handle** to the Contact object on which it intends to operate. It also assumes that the client has acquired a handle to a contact folder to access Contact objects when required.

1.6 Applicability Statement

A client can use this protocol to represent contact information and personal distribution lists 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

The Contact Object Protocol uses the protocols specified in [\[MS-OXCPRPT\]](#) and [\[MS-OXCMSG\]](#) as its primary transport mechanism.

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, as specified in [\[MS-OXCMSG\]](#). The details of how a protocol server operates on Contact objects and Personal Distribution List objects is implementation-dependent, but the results of any such operations MUST be exposed to 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 [\[MS-OXCMSG\]](#). A Contact object or a Personal Distribution List object can 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 object. Unless otherwise specified, each of these properties are optional and they only need to be set when 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.", or "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 is computed by the client according to an implementation-dependent algorithm that uses the 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 client 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 client 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.

The following table shows the format of the **PtypString** property [PidLidFileUnder](#) when all of the properties are present. If a property is not present, then the separator characters surrounding it can be removed by the client.

Computing the value of the [PidLidFileUnder](#) property based on [PidLidFileUnderId](#) property value.

Value of the PidLidFileUnderId property	Meaning of the PidLidFileUnder property
0x00000000	Empty PtypString .
0x00003001	"< PidTagDisplayName >"
0x00003A06	"< PidTagGivenName >"
0x00003A11	"< PidTagSurname >"
0x00003A16	"< PidTagCompanyName >"
0x00008017	"< PidTagSurname >, <space> < PidTagGivenName > <space> < PidTagMiddleName >"
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 >"
0xffffffffd	Specifies that, when displaying the Contact object, the client is to 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	Specifies that, when displaying the Contact object, the client is to choose the appropriate default values (according to the language locale) for PidLidFileUnderId and update PidLidFileUnder to match the choice.
0xfffffffff	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](#). The [PidLidFileUnderList](#) property is set by the client, but it is never used by either the client or the server.

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.

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](#). The following table specifies the properties in each electronic address group.

Group name	Description	Properties in group
Email1	Refers to the group of properties that define the first e-mail address for a contact.	PidLidEmail1DisplayName PidLidEmail1AddressType PidLidEmail1EmailAddress PidLidEmail1OriginalDisplayName PidLidEmail1OriginalEntryId
Email2	Refers to the group of properties that define the second e-mail address for a contact.	PidLidEmail2DisplayName PidLidEmail2AddressType PidLidEmail2EmailAddress PidLidEmail2OriginalDisplayName PidLidEmail2OriginalEntryId
Email3	Refers to the group of properties that define the third e-mail address for a contact.	PidLidEmail3DisplayName PidLidEmail3AddressType PidLidEmail3EmailAddress PidLidEmail3OriginalDisplayName PidLidEmail3OriginalEntryId
Primary Fax	Refers to the group of properties that define the primary fax address for a contact.	PidTagPrimaryFaxNumber PidLidFax1AddressType PidLidFax1EmailAddress PidLidFax1OriginalDisplayName PidLidFax1OriginalEntryId
Business Fax	Refers to the group of properties that define the business fax address for a contact.	PidTagBusinessFaxNumber , PidLidFax2AddressType , PidLidFax2EmailAddress , PidLidFax2OriginalDisplayName PidLidFax2OriginalEntryId
Home Fax	Refers to the group of properties that define the home fax address	PidTagHomeFaxNumber

Group name	Description	Properties in group
	for a contact.	PidLidFax3AddressType PidLidFax3EmailAddress PidLidFax3OriginalDisplayName 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. The client sets the [PidLidAddressBookProviderEmailList](#) and [PidLidAddressBookProviderArrayType](#) and Primary Fax, Business Fax, or Home Fax properties only if the client is able to interpret the value of the [PidTagPrimaryFaxNumber](#), [PidTagBusinessFaxNumber](#), or [PidTagHomeFaxNumber](#) property, respectively, as an actual fax phone number. The server sets the [PidLidAddressBookProviderEmailList](#) and [PidLidAddressBookProviderArrayType](#) properties whenever the FaxNumber property is set. <4>

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; if present, the property value MUST be a valid address type. [RopGetAddressTypes](#) can be used to get a list of the valid address types; a third-party-defined address type is also valid. Address types commonly used by the client and server are "SMTP" and "EX".

2.2.1.2.3 PidLidEmail1EmailAddress, PidLidEmail2EmailAddress, PidLidEmail3EmailAddress

This **PtypString** property specifies the e-mail address of the contact. The format of an e-mail address is determined by its address type. Therefore, the value of this property MUST be appropriate for the address type specified for this e-mail address. Address types are specified by the [PidLidEmail1AddressType](#), [PidLidEmail2AddressType](#), and [PidLidEmail3AddressType](#) properties. The client and server commonly use the "SMTP" and "EX" address types. The format of an "SMTP" e-mail address is specified in [\[RFC821\]](#); the format of an "EX" e-mail address is specified in [\[RFC2247\]](#).

The address type is intended to provide an avenue for various third-party delivery mechanisms to define their own e-mail address format. A Contact object merely stores the address type and the e-mail address; the address type and the e-mail address have no special meaning in the context of this protocol.

2.2.1.2.4 PidLidEmail1OriginalDisplayName, PidLidEmail2OriginalDisplayName, PidLidEmail3OriginalDisplayName

This **PtypString** property SHOULD specify the **Simple Mail Transfer Protocol (SMTP)** e-mail address corresponding to the e-mail address specified in section [2.2.1.2.3](#) for the Contact object. If the value of the [PidLidEmail1AddressType](#), [PidLidEmail2AddressType](#), or [PidLidEmail3AddressType](#) property is "SMTP", then the value of the respective [PidLidEmail1OriginalDisplayName](#), [PidLidEmail2OriginalDisplayName](#), or [PidLidEmail3OriginalDisplayName](#) property SHOULD equal the value of the respective [PidLidEmail1EmailAddress](#), [PidLidEmail2EmailAddress](#), or [PidLidEmail3EmailAddress](#) property; otherwise, the values do not have to be equal. 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, not including the terminating NULL character. There are no other restrictions on the format of this **PtypString**.

2.2.1.2.7 PidLidFax1AddressType, PidLidFax2AddressType, PidLidFax3AddressType

This **PtypString** property specifies the address type of the electronic address. This property, if present, MUST be set to "FAX".

2.2.1.2.8 PidLidFax1EmailAddress, PidLidFax2EmailAddress, PidLidFax3EmailAddress

This **PtypString** property specifies a user-friendly combination of the display name and the corresponding fax number. This 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 specifies the normalized subject for the contact. This property, if present, MUST be set to the same value as [PidTagNormalizedSubject](#), as specified in section [2.2.1.10.1](#).

2.2.1.2.10 PidLidFax1OriginalEntryId, PidLidFax2OriginalEntryId, PidLidFax3OriginalEntryId

This **PtypBinary** property, if present, specifies the one-off EntryID corresponding to this fax address.

2.2.1.2.11 PidLidEmailList

The value of this **PtypMultipleInteger32** property MUST be ignored. <5>

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.

For example, if one of the values in [PidLidAddressBookProviderEmailList](#) is "0x00000000", then [PidLidAddressBookProviderArrayType](#) would 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](#) would 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: Home Address, Work Address, and 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.

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, when one property is set, all of the properties for that address MUST also be set. The following table specifies the properties in each physical address group.

Group name	Properties in group
Home Address	PidTagHomeAddressStreet PidTagHomeAddressCity PidTagHomeAddressStateOrProvince PidTagHomeAddressPostalCode PidTagHomeAddressCountry PidLidHomeAddressCountryCode PidTagHomeAddressPostOfficeBox PidLidHomeAddress
Work Address	PidLidWorkAddressStreet PidLidWorkAddressCity PidLidWorkAddressState PidLidWorkAddressPostalCode PidLidWorkAddressCountry PidLidWorkAddressCountryCode PidLidWorkAddressPostOfficeBox PidLidWorkAddress.
Other Address	PidTagOtherAddressStreet PidTagOtherAddressCity PidTagOtherAddressStateOrProvince PidTagOtherAddressPostalCode PidTagOtherAddressCountry PidLidOtherAddressCountryCode PidTagOtherAddressPostOfficeBox PidLidOtherAddress.
Mailing Address	PidTagStreetAddress PidTagLocality PidTagStateOrProvince PidTagPostalCode PidTagCountry PidLidAddressCountryCode

Group name	Properties in group
	PidTagPostOfficeBox PidTagPostalAddress

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

This **PtypString** property specifies the street portion of the contact's work, home, other, or mailing address. This property can also be used to store the post office box part of the address, if it exists.

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

This **PtypString** property specifies the city or locality portion 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 portion 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) portion 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 portion 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 portion 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 portion of the contact's work, home, other, or mailing address. The client and the server do not use these 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.

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 specified in the following table. If not set, the client 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 all MUST NOT be set.
0x00000001	The Home 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 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 PidTagCompanyMainTelephoneNumber

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 **PtypTime** 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 of the two events is defined for a contact, then [PidLidReferenceEntryId](#) SHOULD be set to the EntryID for this Contact object.

2.2.1.5.1 PidTagBirthday

If present, this **PtypTime** ([\[MS-OXCDATA\]](#), section [2.11.1](#)) property specifies the birthday of the contact, at 0:00 in the client's local time zone. This time is then converted and saved in **Coordinated Universal Time (UTC)** time.

2.2.1.5.2 PidLidBirthdayLocal

If present, this **PtypTime** property specifies the birthday of the contact, at 0:00 in the client's local time zone and it is saved without any time zone conversions.

2.2.1.5.3 PidLidBirthdayEventEntryId

If present, this **PtypBinary** property specifies the object EntryID of an optional **Appointment object** [\[MS-OXOCAL\]](#) that represents the contact's birthday. The Appointment object specified by [PidLidBirthdayEventEntryId](#) MUST be linked to this contact using [PidLidContactLinkEntry](#), [PidLidContactLinkSearchKey](#), and [PidLidContactLinkName](#), as specified in [\[MS-OXCMSG\]](#) section 2.2.1.23.

2.2.1.5.4 PidTagWeddingAnniversary

If present, this **PtypTime** property specifies the wedding anniversary of the contact, at 0:00 in the client's local time zone. This time is then converted and saved in UTC time.

2.2.1.5.5 PidLidWeddingAnniversaryLocal

If present, this **PtypTime** property specifies the wedding anniversary of the contact, at 0:00 in the client's local time zone and it is saved without any time zone conversions.

2.2.1.5.6 PidLidAnniversaryEventEntryId

If present, this **PtypBinary** property specifies the object EntryID of the Appointment object, [\[MS-OXOCAL\]](#) that represents the contact's anniversary. The Appointment object specified by [PidLidAnniversaryEventEntryId](#) MUST be linked to this contact using [PidLidContactLinkEntry](#), [PidLidContactLinkSearchKey](#), and [PidLidContactLinkName](#), as specified in [\[MS-OXCMSG\]](#) section 2.2.1.23.

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 to which the contact belongs.

2.2.1.6.4 PidLidDepartment

This optional **PtypString** property is ignored by the server. The client MUST set this property to an empty string.

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. [<6>](#) The business card format is a collection of information about how to display contact data to the user. The server ignores the business card properties.

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 a contact photo, specified in section [2.2.1.8](#), or a 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 diagram specifies the buffer format of the [PidLidBusinessCardDisplayDefinition](#) property.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
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, as specified in section [2.2.1.7.1.1.1](#).

MinorVersion (1 byte): An 8-bit value that specifies the minor version number, as specified in section [2.2.1.7.1.1.1](#).

TemplateID (1 byte): An 8-bit value that specifies the **display template** to use, as specified in section [2.2.1.7.1.1.2](#).

CountOfFields (1 byte): An 8-bit value that specifies the count of **FieldInfo** structures in the [PidLidBusinessCardDisplayDefinition](#) property.

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, as 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 that 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 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 the value of **CountOfFields**.

ExtraInfo (variable): A byte array that specifies additional information, as specified in section [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. A user agent implementing this protocol SHOULD set the value of **MajorVersion** to "0x03" and SHOULD set the value of **MinorVersion** to "0x00".

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 area will be left aligned, stretching the full height of the card vertically; text fields will appear to the right of the image area.
0x01	The image area will be right aligned, stretching the full height of the card vertically; text fields will appear to the left of the image area.
0x02	The image area will be aligned to the top, stretching the full width of the card horizontally; text fields will appear under the image area.
0x03	The image area will be aligned to the bottom, stretching the full width of the card horizontally; text fields will appear above the image area.
0x04	No image area is included in the card, only text fields are included. PidLidBusinessCardCardPicture SHOULD NOT be set on the Contact object in this case.

Value	Description
0x05	The image area 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 area.

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	Align 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 [2.2.1.8.1](#), or the card picture, as specified in section [2.2.1.7.2](#).

If the value of the **ImageSource** field is 0x00, then the contact photo SHOULD be used; otherwise, the card picture SHOULD be used. If the value of this field is 0x00, then the [PidLidBusinessCardCardPicture](#) property SHOULD NOT exist on the Contact object. This field MUST NOT be set to 0x00 when the value of the [PidLidHasPicture](#) property is FALSE (0x00000000). This field is ignored for text-only cards (when the value of the **TemplateID** field is 0x04).

2.2.1.7.1.1.5 BackgroundColor

A **PtypInteger32** value representing the color of the card background, expressed as 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.

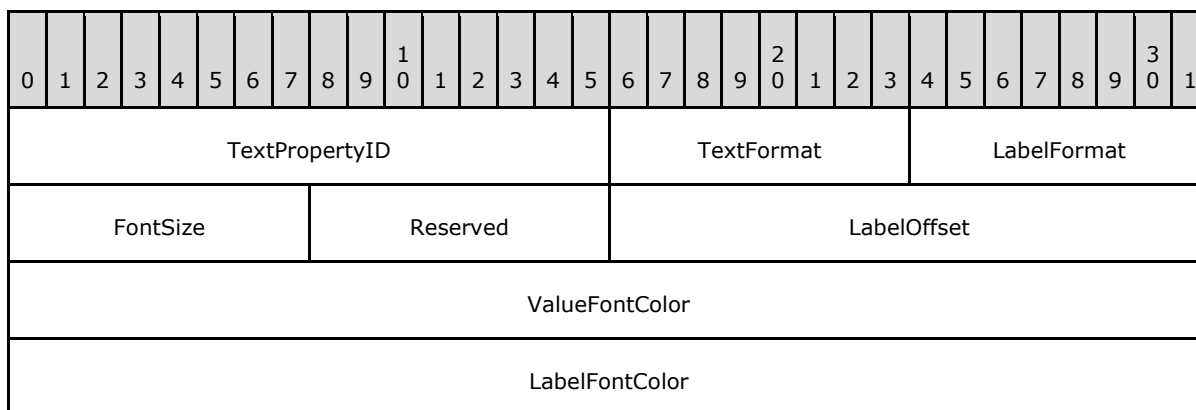
2.2.1.7.1.1.6 ImageArea

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 diagram specifies the buffer format of the **FieldInfo** structure.



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**, as specified in section [2.2.1.7.1.2.4](#).

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, as specified in section [2.2.1.7.1.2.6](#).

LabelFontColor (4 bytes): A **PtypInteger32** value that specifies the color reference code for the label font color, as specified in section [2.2.1.7.1.2.7](#).

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](#)
[PidTagCompanyMainTelephoneNumber](#)
[PidTagHomeTelephoneNumber](#)
[PidTagHome2TelephoneNumber](#)
[PidTagHomeFaxNumber](#)
[PidTagMobileTelephoneNumber](#)
[PidTagAssistantTelephoneNumber](#)
[PidTagOtherTelephoneNumber](#)
[PidTagTtyTddPhoneNumber](#)
[PidTagPrimaryTelephoneNumber](#)
[PidTagPrimaryFaxNumber](#)
[PidTagPagerTelephoneNumber](#)
[PidLidWorkAddress](#)
[PidLidHomeAddress](#)
[PidLidOtherAddress](#)
[PidLidInstantMessagingAddress](#)
[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 diagram are set, the field text is displayed as a 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 diagram 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
					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 to 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 terminating character. The value of **LabelOffset** MUST be less than the value of the **ExtraInfoSize** field, 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 (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.

2.2.1.7.1.2.7 LabelFontColor

LabelFontColor is a color reference code indicating the font color of the label (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.

2.2.1.7.1.3 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 terminating NULL character. Each of these **PtypStrings** SHOULD be referenced by a **LabelOffset** field in one or more **FieldInfo** structures, as specified in section [2.2.1.7.1.2](#). The total size, in bytes, of the **ExtraInfo** field MUST be specified by the value of the **ExtraInfoSize** field.

2.2.1.7.2 PidLidBusinessCardCardPicture

A **PtypBinary** property that contains the image to be used on a business card, whose value MUST be either a **portable network graphics (PNG)** or **.jpg** 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](#)) does 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. [<7>](#)

2.2.1.8.1 PidLidHasPicture

If this **PtypBoolean** property exists and is set to TRUE ("0x00000001"), then the corresponding attachment specified in section [2.2.1.8.2](#) SHOULD exist, and the client SHOULD use it as the contact photo. If this property does not exist, or exists and is FALSE ("0x00000000"), then there is no corresponding attachment.

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 [\[MS-OXCMSG\]](#) section 2.2.2. 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 (0x00000001). There SHOULD be only one attachment with [PidTagAttachmentContactPhoto](#) set to TRUE on a given Contact object. If there is more than one attachment that has this property set, the client is free to pick any of the marked attachments as the contact photo. For more details about [PidTagAttachmentContactPhoto](#), see [\[MS-OXPROPS\]](#) section 2.663.
- The value of the **PtypBinary** [PidTagAttachDataBinary](#) property, which is the contents of the attachment, SHOULD be a stream in .jpg format. The client and server save the contact photo attachment in .jpg format. The client displays the photo if it is in any of the following formats: **BMP, GIF, .jpg, EXIF, PNG, TIFF, enhanced metafile format (EMF), Windows metafile format (WMF), ICO.**
- The value of the **PtypString** [PidTagAttachExtension](#) 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 is equal to the value of the EntryID of the Contact object unless the Contact object is a copy of an earlier original.

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/partner.

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

2.2.1.9.6 PidLidInstantMessagingAddress

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 for the contact, such as 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 **Uniform Resource Locator (URL)** path from which a client can retrieve **free/busy** information for the contact as an iCal file, as specified in [\[MS-OXCICAL\]](#).

2.2.1.9.11 PidTagAccount

This optional **PtypString** property specifies the account name of the contact. This property is not used by either the client or the server.

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 to which the contact's computer is connected. This property is not used by either the client or the server.

2.2.1.9.17 PidTagChildrensNames

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

2.2.1.9.18 PidLidContactCharacterSet

This optional unsigned **PtypInteger32** property specifies the **character set** used for this Contact object. [<8>](#) Clients 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 [PidLidContactCharacterSet](#) property is 0x00000000 or 0x00000001, clients SHOULD treat this

property as not being set. For US English, the client sets [PidLidContactCharacterSet](#) to 0x00000100, denoting a Western character set.

2.2.1.9.19 PidLidAutoLog

This optional **PtypBoolean** property specifies to the client whether 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 the 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, the property **MUST** have six entries, each corresponding to a visible field in the client'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 client SHOULD display the contact's home address.
	0x00000002 or 0x00000000	The client SHOULD display the contact's work address.
	0x00000003	The client SHOULD display the contact's other address.
2	0x00008080	The client SHOULD display Email1.
	0x00008090	The client SHOULD display Email2.

One-based index into the multi-valued property	The value MUST be one of the following	Meaning
	0x000080A0	The client SHOULD display Email3.
3,4,5,6	Property ID 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 client SHOULD display the corresponding property.

2.2.1.9.23 PidTagUserX509Certificate

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

2.2.1.9.24 PidLidBilling

This optional **PtypString** property specifies billing information for the contact.

2.2.1.10 Additional Property Constraints

This protocol specifies additional constraints on the following properties in addition to those 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 is computed by the client according to an implementation-dependent algorithm that uses values of the [PidTagGivenName](#), [PidTagMiddleName](#), [PidTagSurname](#), [PidTagGeneration](#), and [PidTagCompanyName](#) properties. The client 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.

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. Unless otherwise specified, each of these properties are optional and they only need to be set when there is user data that needs to be stored.

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<9> 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<10> 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 lists, 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 [\[MS-OXCDATA\]](#) section 2.2.5.1, 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 15,000 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, <11> it 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 15,000 bytes in size.

2.2.2.2.3 PidLidDistributionListChecksum

This **PtypInteger32** property specifies the 32-bit **cyclic redundancy check (CRC)** polynomial checksum, as specified in [\[ISO/IEC8802-3\]](#), calculated on the value of [PidLidDistributionListMembers](#), as specified in section [3.1.4.2.3.4](#).

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 on the existing value of [PidLidDistributionListMembers](#) and comparing it with the value of the [PidLidDistributionListChecksum](#) property). The server does not set or update this property.

2.2.2.2.4 PidLidDistributionListStream

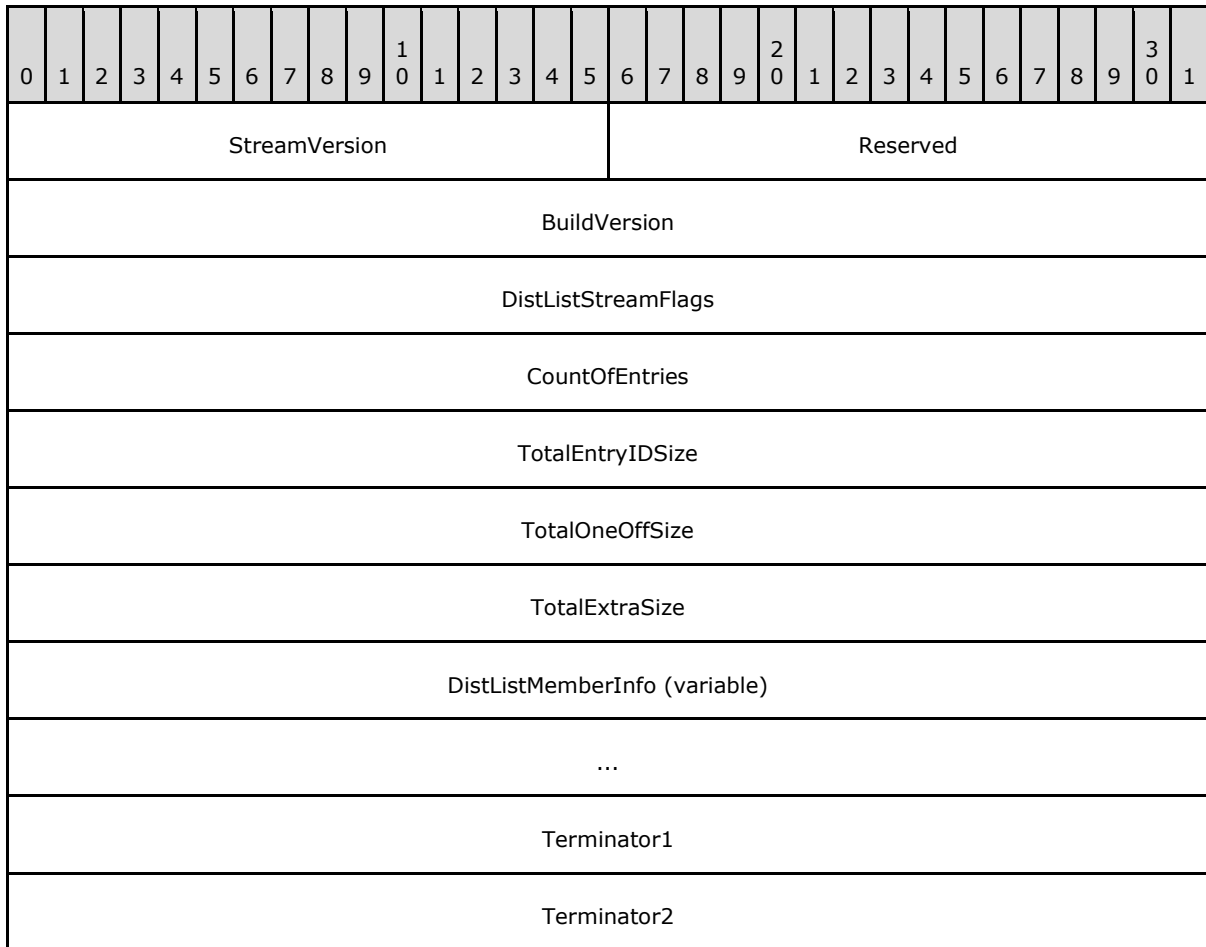
This **PtypBinary** property specifies the list of EntryIDs and one-off EntryIDs corresponding to the members of the personal distribution list. Members of the personal distribution list can be other personal distribution lists, electronic addresses contained in a contact, Global Address List users or distribution lists, or one-off e-mail addresses. The format of each EntryID MUST be as described in section [2.2.2.2.1](#), and the format for each one-off EntryID MUST be as described in section [2.2.2.2.2](#).

[PidLidDistributionListStream](#) is intended to allow distribution lists to grow past the size limits of [PidLidDistributionListMembers](#) and [PidLidDistributionListOneOffMembers](#). This property SHOULD<12> be used if either [PidLidDistributionListMembers](#) or [PidLidDistributionListOneOffMembers](#) would be over 15,000 bytes. If this property is set, [PidLidDistributionListMembers](#), [PidLidDistributionListOneOffMembers](#), and [PidLidDistributionListChecksum](#) SHOULD<13> be ignored.

The server does not use the [PidLidDistributionListStream](#) property.

2.2.2.2.4.1 PidLidDistributionListStream Buffer Format

The following diagram specifies the buffer format of the [PidLidDistributionListStream](#) property.



StreamVersion (2 bytes): MUST be set to 0x0001.

Reserved (2 bytes): MUST be set to "0x0000".

BuildVersion (4 bytes): A 32-bit value that takes the major version number, as specified in section [2.2.1.7.1.1.1](#), and multiplies it by 10,000. The minor version, also specified in section [2.2.1.7.1.1.1](#), is then added to the result.

DistListStreamFlags (4 bytes): This 32-bit value contains bit flags that indicate overall behavior of the personal distribution list, as specified in the table below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Always Use Stream																																	

If the Always Use Stream bit is set then a client MUST save the personal distribution list members in the stream, even if the members would normally fit in [PidLidDistributionListMembers](#) and [PidLidDistributionListOneOffMembers](#).

When modifying the personal distribution list a client MUST preserve any data in the Reserved portion of this value. When creating a new personal distribution list this value MUST set the reserved bits to 0.

CountOfEntries (4 bytes): A 32-bit value that specifies the number of **DistListMemberInfo** structures in the [PidLidDistributionListStream](#).

TotalEntryIDSize (4 bytes): Value representing the sum of the sizes (in bytes) of the EntryIDs stored in each **DistListMemberInfo**.

TotalOneOffSize (4 bytes): Value representing the sum of the sizes (in bytes) of the one-off EntryIDs stored in each **DistListMemberInfo**.

TotalExtraSize (4 bytes): Value representing the sum of the sizes (in bytes) of the **ExtraMemberInfoData** for each member of the distribution list.

DistListMemberInfo (variable): An array of structures that contain information about one member of the Personal Distribution List, as specified in section [2.2.2.2.4.2](#).

Terminator1 (4 bytes): MUST be set to "0x00000000".

Terminator2 (4 bytes): MUST be set to "0x00000000".

2.2.2.2.4.2 DistListMemberInfo Format

The following table specifies the format of each **DistListMemberInfo** structure stored in the [PidLidDistributionListStream](#) buffer.

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
EntryIdSize																															
EntryIdData (variable)																															
...																															
one-offEntryIdSize																															
one-offEntryIdData (variable)																															
...																															
ExtraMemberInfoSize																															
ExtraMemberInfoData (variable)																															
...																															

EntryIdSize (4 bytes): A 32-bit value representing the count of bytes contained in **EntryIdData**.

EntryIdData (variable): A byte array corresponding to the byte array of a **PtypBinary** representing the EntryID of this member of the personal distribution list. Members can be other personal distribution lists, 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 [\[MS-OXCDATA\]](#) section 2.2.5.1, or a **WrappedEntryId**, as specified in section [2.2.2.2.5](#).

one-offEntryIdSize (4 bytes): A 32-bit value denoting the count of bytes contained in **one-offEntryIdData**. If there is no corresponding **one-offEntryID** for this member, this value MUST be set to 0x00000000.

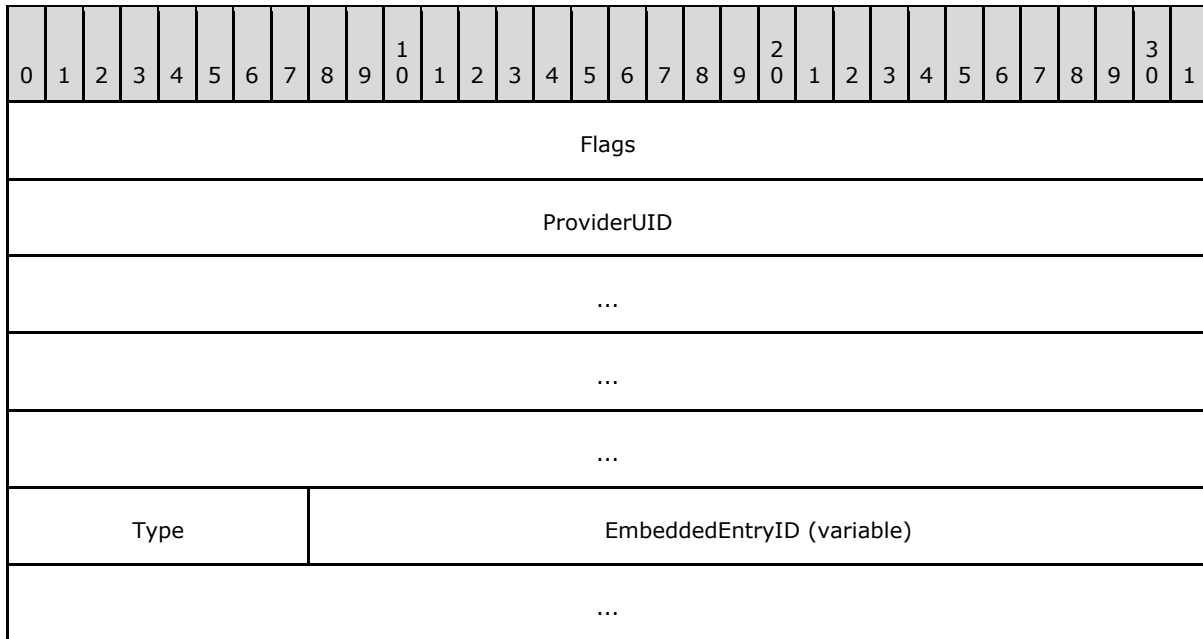
one-offEntryIdData (variable): A byte array corresponding to the byte array of a **PtypBinary** representing the **one-offEntryId** of this member of the personal distribution list. These one-off EntryIDs encapsulate display names and e-mail addresses of the personal distribution list members.

ExtraMemberInfoSize (4 bytes): A 32-bit value denoting the count of bytes contained in the **ExtraMemberInfoData** field. This field MUST be set to 0.

ExtraMemberInfoData (variable): This field is not used by either the client or the server and MUST NOT be present.

2.2.2.2.5 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 diagram.



Flags (4 bytes): MUST be "0x00000000".

ProviderUID (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 EmbeddedEntryID field MUST be a one-off EntryID as specified in [MS-OXCDATA] section 2.2.5.1. A value of 3 designates a Contact object EntryID; the EmbeddedEntryID field MUST be the message EntryID (as specified in [MS-OXCDATA] section 2.2.4.2) of a message Contact object. A value of 4 designates a Personal Distribution List object EntryID; the EmbeddedEntryID field MUST be the message EntryID (as specified in [MS-OXCDATA] section 2.2.4.2) of a Personal Distribution List object. A value of 5 designates a Global Address List local mail user EntryID; the EmbeddedEntryID field MUST be the Global Address List EntryID (as specified in [MS-OXCDATA] section 2.2.5.2) of a user in the Global Address List. A value of 6 designates a Global Address List distribution list EntryID; the

Bit Mask	Description
	EmbeddedEntryID field MUST be the Global Address List EntryID (as specified in [MS-OXCDATA] section 2.2.5.2) 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 Email1 address) ▪ 5 (denoting an Email2 address) ▪ 6 (denoting an Email3 address) <p>Note that this value MUST NOT be set to 7.</p>
0x80	If the embedded EntryID is a one-off EntryID, this bit MUST NOT be set. Otherwise, this bit MUST be set.

EmbeddedEntryID (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 [<14>](#) 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 [\[MS-OXCFOLD\]](#) that have the value of the [PidTagContainerClass](#) **PtypString** property set to "IPF.Contact", or a **PtypString** prefixed with "IPF.Contact.". For more details about [PidTagContainerClass](#), see [\[MS-OXPROPS\]](#) section 2.714. In all other respects, they conform to the folder syntax and functionality specified by [\[MS-OXCFOLD\]](#).

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

3 Protocol Details

3.1 Common 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 of 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 of the properties synchronized with one another. There is also one property that specifies which of the Physical Address subobjects 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 specifies an optional image of the person or entity represented by the Contact object. This image exists independently of the **Business Card** object, but can be used by it.

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 client creates a Message object, as specified in [\[MS-OXCMSG\]](#) section 3.1.4.2, sets properties in accordance with the requirements in section [2.2.1](#) of this specification and [\[MS-OXCPRPT\]](#), and saves the resulting Message object as described in [\[MS-OXCMSG\]](#) section 3.1.4.3.

3.1.4.1.2 Deleting a Contact

Contact objects have no special semantics in relation to deletion beyond what is specified in [\[MS-OXCFOOLD\]](#).

3.1.4.1.3 Modifying a Contact

When modifying a Contact object, the client opens a Message object as specified in [\[MS-OXCMSG\]](#) section 3.1.4.1, modifies any of the properties in accordance with the requirements in [\[MS-OXCPRPT\]](#) and in section [2](#) and sections [3.1.4.1.3.1](#) through [3.1.4.1.3.7](#) of this specification, and saves the Message object as described in [\[MS-OXCMSG\]](#) section 3.1.4.3.

3.1.4.1.3.1 Modifying a Contact Name Property

Contact name properties **MUST** be modified at the same time in order to preserve their relationship. The following table specifies the properties that **SHOULD** be updated when another property is changed (according to the definition of each property in section [2.2.1](#) of this specification). More than one **condition** can be met at the same time. Not all properties will need to be updated every time. Clients use the specifications in section [2.2.1.3](#) to determine whether an update is required.

Changed property	Conditions	Properties that SHOULD 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
PidTagDisplayNamePrefix	None.	PidTagDisplayName
PidTagSurname or PidTagMiddleName or PidTagGivenName	None.	PidTagDisplayName , PidTagNormalizedSubject , PidTagInitials
PidTagDisplayName	None.	PidTagDisplayNamePrefix , PidTagGivenName , PidTagMiddleName , PidTagSurname , PidTagNormalizedSubject , PidTagGeneration

The client SHOULD also update [PidLidFax1EmailAddress](#), [PidLidFax2EmailAddress](#), or [PidLidFax3EmailAddress](#) 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 specification). Not all properties will need to be updated every time. Clients use the specifications in section 2.2.1.3 to determine whether an update is required. More than one condition can be met at the same time.

Changed property	Properties that SHOULD be updated
PidLidPostalAddressId	PidTagPostalAddress , PidTagStreetAddress , PidTagLocality , PidTagStateOrProvince , PidTagPostalCode , PidTagCountry , PidLidAddressCountryCode
PidLidWorkAddress	PidLidWorkAddressStreet , PidLidWorkAddressCity , PidLidWorkAddressState , PidLidWorkAddressPostalCode , PidLidWorkAddressCountry , PidLidWorkAddressCountryCode
PidLidHomeAddress	PidTagHomeAddressStreet , PidTagHomeAddressCity , PidTagHomeAddressStateOrProvince , PidTagHomeAddressPostalCode , PidTagHomeAddressCountry , PidLidHomeAddressCountryCode
PidLidOtherAddress	PidTagOtherAddressStreet , PidTagOtherAddressCity , PidTagOtherAddressStateOrProvince , PidTagOtherAddressPostalCode , PidTagOtherAddressCountry , PidLidOtherAddressCountryCode
PidTagHomeAddressStreet	PidLidHomeAddress

Changed property	Properties that SHOULD be updated
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
PidLidOtherAddressCountryCode	PidLidOtherAddress

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

If the home address, business address, or other address is updated, and [PidLidPostalAddressId](#) 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 E-mail Address Property

Electronic address properties MUST be modified at the same time in order to preserve their relationship. The following table specifies the properties that SHOULD be updated when another property is changed (according to the definition of each property in section [2.2.1](#) of this specification). More than one condition can be met at the same time. The table includes property names only for the properties in the "Email1" group (see section [2.2.1.2](#)), but the same logic also applies to the properties in the "Email2" and "Email3" groups. Not all properties need to be updated every time. Clients use the specifications defined in section [2.2.1.2](#) to determine whether an update is required.

Changed property	Conditions	Properties that SHOULD 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
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 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 to be updated or defined when another property is changed as specified in section [2.2.1.5](#).

Changed property	Conditions	Properties to be Updated or Defined
PidTagBirthday	None.	PidLidReferenceEntryId , PidLidBirthdayEventEntryId , PidLidBirthdayLocal See the following paragraph for more details.
PidLidBirthdayLocal	None.	PidLidReferenceEntryId , PidLidBirthdayEventEntryId , PidTagBirthday See the following paragraph for more details.
PidTagWeddingAnniversary	None.	PidLidReferenceEntryId , PidLidAnniversaryEventEntryId , PidLidWeddingAnniversaryLocal See the following paragraph for more details.
PidLidWeddingAnniversaryLocal	None.	PidLidReferenceEntryId , PidLidAnniversaryEventEntryId , PidTagWeddingAnniversary See the following paragraph for more details.
PidLidBirthdayEventEntryId	None.	PidTagBirthday , PidLidBirthdayLocal
PidLidAnniversaryEventEntryId	None.	PidTagWeddingAnniversary , PidLidWeddingAnniversaryLocal
PidTagNormalizedSubject	PidLidBirthdayEventEntryId 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 or PidLidAnniversaryEventEntryId is set.	Update the value of the PidLidPrivate property on the corresponding Appointment object to match the one on the contact. For more details about PidLidPrivate , see [MS-OXPROPS] section 2.208.

When the property [PidTagBirthday](#) or [PidLidBirthdayLocal](#) 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](#), [PidLidContactLinkSearchKey](#), and [PidLidContactLinkName](#), as specified in [\[MS-OXCMSG\]](#) section 2.2.1.23.

When the property [PidTagWeddingAnniversary](#) or [PidLidWeddingAnniversaryLocal](#) 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](#), [PidLidContactLinkSearchKey](#), and [PidLidContactLinkName](#), as specified in [\[MS-OXCMSG\]](#) section 2.2.1.23.

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 2.2.1.7.1.1.4 .
PidLidBusinessCardDisplayDefinition	When the ImageSource section of the buffer changes.	PidLidBusinessCardCardPicture MUST be added or removed.

3.1.4.1.3.7 Modifying a Contact Photo Property

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

Changed property or attachment	Conditions	Properties that MUST be updated
Contact Photo Attachment (see section 2.2.1.8.2)	On removal or addition.	PidLidHasPicture MUST be updated to account for the contact photo (see section 2.2.1.8.1).

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 Personal Distribution List object, the client creates a Message object as specified in [\[MS-OXCMSG\]](#) section 3.1.4.2, sets properties in accordance with the requirements in [\[MS-OXCPRPT\]](#) and in section [2.2.2](#) of this specification, and saves the resulting Message object as described in [\[MS-OXCMSG\]](#) section 3.1.4.3.

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 [\[MS-OXCFCM\]](#).

3.1.4.2.3 Modifying a Personal Distribution List

When modifying a Personal Distribution List object, the client opens a Message object as specified in [\[MS-OXCMSG\]](#) section 3.1.4.1, modifies any of the properties in accordance with the requirements in [\[MS-OXCPRPT\]](#) and in section [2](#) and sections [3.1.4.2.3.1](#) through [3.1.4.2.3.4](#) of this specification, and saves the Message object as described in [\[MS-OXCMSG\]](#) section 3.1.4.3.

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 client MUST be able to handle circular references.

3.1.4.2.3.1 Naming a Personal Distribution List

When the client 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 object.
PidLidDistributionListName	The name of the Personal Distribution List object.
PidLidFileUnder	The name of the Personal Distribution List object.
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 List objects SHOULD NOT contain duplicate entries for the same members. Whenever a member is added to a personal distribution list, the client SHOULD check that the new member isn't already in the list. Then the client 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 client 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 client 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 client 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 [PidLidDistributionListChecksum](#) property SHOULD be updated. <15> The checksum is calculated by iterating through each byte of each **PtypBinary** value in the multi-valued [PidLidDistributionListMembers](#) property, as specified by the following algorithm.

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

This algorithm uses the IEEE 802.3 CRC polynomial $0xEDB88320$ ($x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$) with a seed value of $0x00000000$. For more details about the polynomial, see [\[ISO/IEC8802-3\]](#) section 3.2.8.

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), and 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
PidLidContactCharacterSet	{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 can return the following property IDs in response to the [RopGetPropertyIdsFromNames](#) request. The actual property IDs are at the discretion of the server.

Property	Property ID
PidLidFileUnderList	0x827b
PidLidAutoLog	0x8280
PidLidAddressBookProviderEmailList	0x81d4
PidLidAddressBookProviderArrayType	0x81d5
PidLidFileUnder	0x8016
PidLidFileUnderId	0x81da
PidLidContactCharacterSet	0x8286
PidLidEmail1DisplayName	0x 8013

Property	Property ID
PidLidEmail1AddressType	0x81ce
PidLidEmail1EmailAddress	0x 8010
PidLidEmail1OriginalDisplayName	0x801f
PidLidEmail1OriginalEntryId	0x81cf

To create a Contact object, the client uses [RopCreateMessage](#) ([MS-OXCMSG] section 2.2.3.2). 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](#) ([MS-OXCPRPT] section 2.2.5) 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 lp1: {0x00008017; 0x00008037; 0x00003a16; 0x00008019; 0x00008018}

Property	Property type	Data	Meaning
PidLidAddressBookProvider EmailList	PtypMultipleInteger32	01 00 00 00 00 00 00 00	cValues: 0x00000001 lp1: {0x00000000}
PidLidAddressBookProvider ArrayType	PtypInteger32	0x0000 0001	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	0x0000 8017	PidLidFileUnder is "<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
PidTagBusinessTelephoneN umber	PtypString	35 00 35 00 35 00 2d 00 31 00 32 00 33 00 34 00 00	"555-1234"

Property	Property type	Data	Meaning
PidLidContactCharacterSet	PtypInteger32	0x0000 0100	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"
PidLidEmail1DisplayName	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	"user10 (user10@szfkuk- dom.example.com) "

Property	Property type	Data	Meaning
		29 00 00 00	
PidLidEmail1AddressType	PtypString	45 00 58 00 00 00	"EX"
PidLidEmail1EmailAddress	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 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	"/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipient s/cn=user10"

Property	Property type	Data	Meaning
		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	
PidLidEmail1OriginalDisplay Name	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	"user10@szfkuk-dom.example.com"

Property	Property type	Data	Meaning
		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	
PidLidEmail1OriginalEntryId	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	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.

Property	Property type	Data	Meaning
		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	
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 ... (remai nder of proper ty exclud ed for brevit	Size: 5350 bytesR.. LZFub.~.`n g102f5.d .rcp.... 2..`c.Df 3150B7.. stsh.pbt

Property	Property type	Data	Meaning
		y)	
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	""
PidTagSubject	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 his or her changes, the client uses **RopSaveChangesMessage** to commit the properties on the server.

The values of some properties will change during the execution of **RopSaveChangesMessage**; 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](#) ([\[MS-OXCPRPT\]](#) section 2.2.12).

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 can return the following property IDs in response to the [RopGetPropertyIdsFromNames](#) request. The actual property IDs are at the discretion of the server.

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](#) ([MS-OXCMSG] section 2.2.3.2). 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](#) ([MS-OXCPRPT] section 2.2.5) 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"

Property	Property type	Data	Meaning
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	0x0xd4b0223a	-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	2 binary large objects (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.....

Property	Property type	Data	Meaning
		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	
PidLidDistributionListOneOffMembers	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 ac 00 00 00 00 00 81 2b 1f a4 be a3 10 19 9d 6e 00 dd	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: 172 bytes+..n.. ..T..... u.s.e.r. 1.0. .(. u.s.e.r. 1.0.@.s. z.f.k.u. k.-.d.o. m...e.x. a.m.p.l. e...c.o.

Property	Property type	Data	Meaning
		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	m.)...S. M.T.P... u.s.e.r. l.O.@.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	""
PidTagSubject	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 his or her changes, the client uses **RopSaveChangesMessage** to commit the properties on the server.

The values of some properties will change during the execution of **RopSaveChangesMessage**; 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. 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: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Office Outlook® 2003
- Microsoft® Exchange Server 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Exchange Server 2007
- Microsoft® Outlook® 2010
- Microsoft® Exchange Server 2010
- Microsoft® Office Communicator 2007 R2
- Microsoft® Communicator 2010

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

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

<1> [Section 2.2](#): Outlook 2003, Outlook 2007, and Outlook 2010 set the following properties regardless of user input; their values have no meaning in the context of this protocol: [PidLidAgingDontAgeMe](#), [PidLidCurrentVersion](#), [PidLidCurrentVersionName](#), [PidLidValidFlagStringProof](#), [PidTagAlternateRecipientAllowed](#), [PidTagClientSubmitTime](#), [PidTagDeleteAfterSubmit](#), [PidTagImportance](#), [PidTagMessageDeliveryTime](#), [PidTagMessageLocaleId](#), [PidTagOriginatorDeliveryReportRequested](#), [PidLidSideEffects](#), [PidTagPriority](#), [PidTagReadReceiptRequested](#), and [PidTagSensitivity](#).

<2> [Section 2.2](#): Outlook 2007 and Outlook 2010 set 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, Outlook 2007, and Outlook 2010 set the following properties regardless of user input; their values have no meaning in the context of this protocol: [PidLidEmail1RichTextFormat](#), [PidLidEmail2RichTextFormat](#), [PidLidEmail3RichTextFormat](#), and [PidLidReferredBy](#).

<4> [Section 2.2.1.2](#): Exchange 2003 sets the [PidLidAddressBookProviderArrayType](#) property to 0x00000000 and never sets the [PidLidAddressBookProviderEmailList](#) property unless an e-mail address is defined for the contact.

<5> [Section 2.2.1.2.11](#): Outlook 2003, Outlook 2007, and Outlook 2010 set this multi-valued LONG property to contain the three values "0x00008080", "0x00008090", and "0x000080A0" whenever

one e-mail address is defined. Exchange 2003, Exchange 2007, and Exchange 2010 do not set this property.

<6> [Section 2.2.1.7](#): The business card properties are not used by Outlook 2003.

<7> [Section 2.2.1.8](#): Contact Photo properties are not used by Exchange 2003.

<8> [Section 2.2.1.9.18](#): Exchange 2003 and Exchange 2007 do not set or use this property.

<9> [Section 2.2.2.1.2](#): For a newly-created distribution list without a name at all, Outlook 2003, Outlook 2007, and Outlook 2010 do not set the property [PidLidDistributionListName](#) to any value. In the same scenario, Outlook 2003, Outlook 2007, and Outlook 2010 set [PidTagDisplayName](#) to the string containing just a terminating NULL character. In every other case, Outlook 2003, Outlook 2007, and Outlook 2010 keep these two properties the same. Exchange 2003, Exchange 2007, and Exchange 2010 never set this property.

<10> [Section 2.2.2.1.4](#): Exchange 2007 and Exchange 2010 set this property to 0 when it creates a distribution list.

<11> [Section 2.2.2.2.2](#): Exchange 2003 does not set or update this property.

<12> [Section 2.2.2.2.4](#): Outlook 2003 does not use the [PidLidDistributionListStream](#) property.

<13> [Section 2.2.2.2.4](#): If the [PidLidDistributionListStream](#) property is set, Outlook 2003 does not ignore the [PidLidDistributionListMembers](#), [PidLidDistributionListOneOffMembers](#), and [PidLidDistributionListChecksum](#) properties.

<14> [Section 2.2.2.3.1](#): Exchange 2003, Exchange 2007, and Exchange 2010 do not set this property.

<15> [Section 3.1.4.2.3.4](#): Exchange 2003, Exchange 2007, and Exchange 2010 do not update this property.

7 Change Tracking

This section identifies changes that were made to the [MS-OXOCNTC] protocol document between the May 2010 and August 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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

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

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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

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

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.1 Glossary	55658 Added the term "handle", removed the term "journal".	N	Content update.
1.2.1 Normative References	55751 Moved [MS-OXGLOS] from Normative References section to Informative References section.	N	Content update.
1.2.1 Normative References	57633 Added [RFC821] to list of normative references.	N	Content update.

8 Index

A

Abstract data model
[client](#) 43
[server](#) 43
[Applicability](#) 11

C

[Change tracking](#) 69
Client
[abstract data model](#) 43

D

Data model – abstract
[client](#) 43
[server](#) 43
[DistListMemberInfo packet](#) 38

E

[Examples - overview](#) 52

F

[FieldInfo packet](#) 28

G

[Glossary](#) 8

I

[Implementer - security considerations](#) 66
[Introduction](#) 8

M

Messages
[overview](#) 12

N

[Normative references](#) 9

O

[Overview \(synopsis\)](#) 10

P

[PidLidBusinessCardDisplayDefinition packet](#) 25
[PidLidDistributionListStream Buffer packet](#) 37
[Preconditions](#) 11
[Prerequisites](#) 11
[Product behavior](#) 67

R

References
[normative](#) 9
[Relationship to other protocols](#) 10

S

Security
[implementer considerations](#) 66
[overview](#) 66
Server
[abstract data model](#) 43

T

[Tracking changes](#) 69

W

[Wrapped EntryId packet](#) 40