[MS-OXCEXT]:

Client Extension Message Object Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
4/27/2012	0.1	New	Released new document.
7/16/2012	0.2	Minor	Clarified the meaning of the technical content.
10/8/2012	1.0	Major	Significantly changed the technical content.
2/11/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	1.1	Minor	Clarified the meaning of the technical content.
3/16/2015	2.0	Major	Significantly changed the technical content.
5/26/2015	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2015	2.0	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	2.0	None	No changes to the meaning, language, or formatting of the technical content.
7/24/2018	3.0	Major	Significantly changed the technical content.
10/1/2018	4.0	Major	Significantly changed the technical content.
12/11/2018	4.1	Minor	Clarified the meaning of the technical content.
4/22/2021	5.0	Major	Significantly changed the technical content.
8/17/2021	6.0	Major	Significantly changed the technical content.
2/15/2022	6.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Client Extension Message Object Protocol allows clients to access **mail add-in** data stored in a **mailbox**.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

big-endian: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

Calendar object: A **Message object** that represents an event, which can be a one-time event or a recurring event. The Calendar object includes properties that specify event details such as description, organizer, date and time, and status.

conversation: A single representation of a send/response series of email messages. A conversation appears in the Inbox as one unit and allows the user to view and read the series of related email messages in a single effort.

dictionary: A collection of key/value pairs. Each pair consists of a unique key and an associated value. Values in the dictionary are retrieved by providing a key for which the dictionary returns the associated value.

Email object: A **Message object** that represents an email message in a message store and adheres to the property descriptions that are described in in [MS-OXOMSG].

FAI contents table: A table of folder associated information (FAI) Message objects that are stored in a Folder object.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

GUIDString: A **GUID** in the form of an ASCII or Unicode string, consisting of one group of 8 hexadecimal digits, followed by three groups of 4 hexadecimal digits each, followed by one group of 12 hexadecimal digits. It is the standard representation of a GUID, as described in [RFC4122] section 3. For example, "6B29FC40-CA47-1067-B31D-00DD010662DA". Unlike a curly braced GUID string, a GUIDString is not enclosed in braces.

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC7159]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

mail add-in: An Office Add-in that enhances an email or appointment item.

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

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

named property: A property that is identified by both a GUID and either a string name or a 32-bit identifier.

property name: A string that, in combination with a **property set**, identifies a **named property**.

property set: A set of attributes, identified by a **GUID**. Granting access to a property set grants access to all the attributes in the set.

property tag: A 32-bit value that contains a property type and a property ID. The low-order 16 bits represent the property type. The high-order 16 bits represent the property ID.

Recurring Calendar object: A **Calendar object** that describes an event that repeats according to a recurrence pattern.

remote operation (ROP): An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

remote procedure call (RPC): A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [C706].

ROP response: See ROP response buffer.

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

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

XML document: A document object that is well formed, as described in [XML10/5], and might be valid. An XML document has a logical structure that is composed of declarations, elements, comments, character references, and processing instructions. It also has a physical structure that is composed of entities, starting with the root, or document, entity.

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

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

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

1.2 References

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

1.2.1 Normative References

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

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-OWEMXML] Microsoft Corporation, "Office Web Extensibility Manifest Format".

[MS-OXCDATA] Microsoft Corporation, "Data Structures".

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

[MS-OXCNOTIF] Microsoft Corporation, "Core Notifications Protocol".

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

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol".

[MS-OXCTABL] Microsoft Corporation, "Table Object Protocol".

[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".

[MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List".

[MS-OXWSCEXT] Microsoft Corporation, "Client Extension Web Service Protocol".

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

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

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

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/

1.2.2 Informative References

[MS-OWEXML] Microsoft Corporation, "Office Web Extensibility Extensions to Office Open XML Structure Specification".

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

[MSDN-JavaScriptApiOffice] Microsoft Corporation, "JavaScript API for Office", http://msdn.microsoft.com/en-us/library/fp142185

1.3 Overview

This protocol enables email clients that use **remote procedure call (RPC)** to access user **mailboxes**, as described in [MS-OXCMSG], to read and write data that is used to support mail add-ins. A typical scenario for using this protocol is an email client that supports mail add-ins that extend email or meeting requests, as described in [MS-OWEXML].

This protocol defines the following:

- A method for clients to request notification of changes to the installed and enabled mail add-ins for the mailbox.
- A method for clients to derive a web services item identifier, as described in [MS-OXWSCORE] section 2.2.4.25, for a **Message object**.
- The location and format of **mail add-in**–specific configuration data.
- The location and format of mail add-in–specific custom properties on Message objects.
- The location and format of known entity data on Message objects, as described in [MS-OWEMXML].

1.4 Relationship to Other Protocols

This protocol uses the **remote operations (ROPs)** described in the Message and Attachment Object Protocol, described in [MS-OXCMSG], and in the Property and Stream Object Protocol, described in [MS-OXCPRPT], to access data contained in **Message objects** and configuration data, as described in [MS-OXOCFG]. This protocol also uses the Core Notifications Protocol described in [MS-OXCNOTIF] to register for notifications.

This protocol is used by clients that implement support for mail add-ins, as described in [MS-OWEXML] and the Office Web Extensibility Manifest Format, as described in [MS-OWEMXML].

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is designed to enable email clients that use **RPC** to access user **mailboxes** on the server to implement support for mail add-ins, as described in [MS-OWEXML]. This protocol provides the locations and formats of data stored in user mailboxes required to support mail add-ins.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol uses the protocols specified in [MS-OXCMSG] and [MS-OXCPRPT] as its transport mechanism.

2.2 Message Syntax

2.2.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [XMLNS]. Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1/2]

2.2.2 Known Entity Properties

The following properties contain information about known entities, as specified in [MS-OWEMXML], contained in **Message objects**.

- **PidNameExtractedAddresses** (section 2.2.2.1)
- **PidNameExtractedContacts** (section 2.2.2.2)
- **PidNameExtractedEmails** (section 2.2.2.3)
- **PidNameExtractedMeetings** (section 2.2.2.4)
- **PidNameExtractedPhones** (section 2.2.2.5)
- **PidNameExtractedTasks** (section 2.2.2.6)
- **PidNameExtractedUrls** (section 2.2.2.7)

2.2.2.1 PidNameExtractedAddresses Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedAddresses** property ([MS-OXPROPS] section 2.434) on a **Message object** contains an **XML document** with a single **AddressSet** element, as specified in section 2.2.3.1.3.

2.2.2.2 PidNameExtractedContacts Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedContacts** property ([MS-OXPROPS] section 2.435) on a **Message object** contains an **XML document** with a single **ContactSet** element, as specified in section 2.2.3.1.10.

2.2.2.3 PidNameExtractedEmails Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedEmails** property ([MS-OXPROPS] section 2.436) on a **Message object** contains an **XML document** with a single **EmailSet** element, as specified in section 2.2.3.1.14.

2.2.2.4 PidNameExtractedMeetings Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedMeetings** property ([MS-OXPROPS] section 2.437) on a **Message object** contains an **XML document** with a single **MeetingSet** element, as specified in section 2.2.3.1.20.

2.2.2.5 PidNameExtractedPhones Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedPhones** property ([MS-OXPROPS] section 2.438) on a **Message object** contains an **XML document** with a single **PhoneSet** element, as specified in section 2.2.3.1.27.

2.2.2.6 PidNameExtractedTasks Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedTasks** property ([MS-OXPROPS] section 2.439) on a **Message object** contains an **XML document** with a single **TaskSet** element, as specified in section 2.2.3.1.32.

2.2.2.7 PidNameExtractedUrls Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The value of the **PidNameExtractedUrls** ([MS-OXPROPS] section 2.440) property on a **Message object** contains an **XML document** with a single **UrlSet** element, as specified in section 2.2.3.1.36.

2.2.3 Known Entity XML

The properties specified in section 2.2.2.1 through section 2.2.2.7 contain **XML documents** that use the elements, complex types, simple types, and attributes specified in section 2.2.3.1 through section 2.2.3.4.6.2.

2.2.3.1 Elements

The set of common **XML schema** element definitions defined by this specification is summarized in the following table.

Element	Description
Address (section 2.2.3.1.1)	Specifies a postal or street address.
Addresses (section 2.2.3.1.2)	Specifies a list of postal or street addresses.
AddressSet (section 2.2.3.1.3)	Specifies a set of address known entities.

Element	Description
Assignees (section 2.2.3.1.4)	Specifies a set of assignees for a task.
Attendees (section 2.2.3.1.5)	Specifies a set of attendees for a meeting.
Business (section 2.2.3.1.6)	Specifies information about a business.
BusinessString (section 2.2.3.1.7)	Specifies the name of a business.
Contact (section 2.2.3.1.8)	Specifies information about a contact.
Contacts (section 2.2.3.1.9)	Specifies a list of contacts.
ContactSet (section 2.2.3.1.10)	Specifies a set of contact known entities.
ContactString (section 2.2.3.1.11)	Specifies the name of a contact.
Email (section 2.2.3.1.12)	Specifies information about an email address.
Emails (section 2.2.3.1.13)	Specifies a list of email addresses.
EmailSet (section 2.2.3.1.14)	Specifies a set of email address known entities.
EmailString (section 2.2.3.1.15)	Specifies an email address.
EmailUser (section 2.2.3.1.16)	Specifies information about an email user.
EndTime (section 2.2.3.1.17)	Specifies the end date and time of a meeting.
Meeting (section 2.2.3.1.18)	Specifies information about a meeting suggestion.
Meetings (section 2.2.3.1.19)	Specifies a list of meeting suggestions.
MeetingSet (section 2.2.3.1.20)	Specifies a set of meeting known entities.
MeetingString (section 2.2.3.1.21)	Specifies a string that represents a meeting suggestion.
OriginalPhoneString (section 2.2.3.1.22)	Specifies a phone number before normalization.
Person (section 2.2.3.1.23)	Specifies information about a person.
PersonString (section 2.2.3.1.24)	Specifies the name of a person.
Phone (section 2.2.3.1.25)	Specifies information about a phone number.
Phones (section 2.2.3.1.26)	Specifies a list of phone numbers.
PhoneSet (section 2.2.3.1.27)	Specifies a set of phone number known entities.
PhoneString (section 2.2.3.1.28)	Specifies a normalized phone number.
StartTime (section 2.2.3.1.29)	Specifies the start date and time of a meeting.
Task (section 2.2.3.1.30)	Specifies information about a task suggestion.
Tasks (section 2.2.3.1.31)	Specifies a list of task suggestions.
TaskSet (section 2.2.3.1.32)	Specifies a set of task known entities.
TaskString (section 2.2.3.1.33)	Specifies a string that represents a task suggestion.
Url (section 2.2.3.1.34)	Specifies information about a **URL**.

Element	Description
Urls (section 2.2.3.1.35)	Specifies a list of URLs.
UrlSet (section 2.2.3.1.36)	Specifies a set of URL known entities.
UrlString (section 2.2.3.1.37)	Specifies a URL.
Version (section 2.2.3.1.38)	Specifies the version of the XML schema that applies to the **XML document**.

2.2.3.1.1 Address Element

Type: **Address** (section 2.2.3.2.1)

The **Address** element contains a string that represents a postal or street address. It is an optional child element of the **Addresses** element specified in section 2.2.3.1.2.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Address"
    nillable="true" type="Address" />
```

2.2.3.1.2 Addresses Element

Type: **ArrayOfAddress** (section 2.2.3.2.3)

The **Addresses** element contains a list of postal or street addresses. It is an optional child element of the **AddressSet** element specified in section 2.2.3.1.3 and the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Addresses" type="ArrayOfAddress" />
```

2.2.3.1.3 AddressSet Element

Type: **AddressSet** (section 2.2.3.2.2)

The **AddressSet** element is the root element for the **XML document** contained in the **PidNameExtractedAddresses** property specified in section 2.2.2.1. This element contains the known entities that represent addresses, as specified in [MS-OWEMXML].

```
<xs:element name="AddressSet" nillable="true" type="AddressSet" />
```

2.2.3.1.4 Assignees Element

Type: **ArrayOfEmailUser** (section 2.2.3.2.6)

The **Assignees** element contains a list of assignees for a task. It is an optional child element of the **Task** element specified in section 2.2.3.1.30.

```
<xs:element minOccurs="0" maxOccurs="1" name="Assignees" type="ArrayOfEmailUser" />
```

2.2.3.1.5 Attendees Element

Type: **ArrayOfEmailUser** (section 2.2.3.2.6)

The **Attendees** element contains a list of attendees for a meeting. It is an optional child element of the **Meeting** element specified in section 2.2.3.1.18.

```
<xs:element minOccurs="0" maxOccurs="1" name="Attendees" type="ArrayOfEmailUser" />
```

2.2.3.1.6 Business Element

Type: **Business** (section 2.2.3.2.11)

The **Business** element contains elements that represent the name of a business. It is an optional child element of the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Business" type="Business" />
```

2.2.3.1.7 BusinessString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **BusinessString** element contains a string that represents the name of a business. It is an optional child element of the **Business** element specified in section 2.2.3.1.6.

```
<xs:element minOccurs="0" maxOccurs="1" name="BusinessString" type="xs:string" />
```

2.2.3.1.8 Contact Element

Type: **Contact** (section 2.2.3.2.12)

The **Contact** element contains information about a contact, such as name, phone number, or email address. It is an optional child element of the **Contacts** element specified in section 2.2.3.1.9.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Contact"
    nillable="true" type="Contact" />
```

2.2.3.1.9 Contacts Element

Type: **ArrayOfContact** (section 2.2.3.2.4)

The **Contacts** element contains a list of contacts. It is an optional child element of the **ContactSet** element specified in section 2.2.3.1.10.

```
<xs:element minOccurs="0" maxOccurs="1" name="Contacts" type="ArrayOfContact" />
```

2.2.3.1.10 ContactSet Element

Type: **ContactSet** (section 2.2.3.2.13)

The **ContactSet** element is the root element for the **XML document** contained in the **PidNameExtractedContacts** property specified in section 2.2.2.2. This element contains the known entities that represent contacts, as specified in [MS-OWEMXML].

```
<xs:element name="ContactSet" nillable="true" type="ContactSet" />
```

2.2.3.1.11 ContactString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **ContactString** element contains a string that represents a contact. It is an optional child element of the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="ContactString" type="xs:string" />
```

2.2.3.1.12 Email Element

Type: **Email** (section 2.2.3.2.14)

The **Email** element contains elements that represent an email address. It is an optional child element of the **Emails** element specified in section 2.2.3.1.13.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Email" nillable="true" type="Email" />
```

2.2.3.1.13 Emails Element

Type: **ArrayOfEmail** (section 2.2.3.2.5)

The **Emails** element contains a list of email addresses. It is an optional child element of the **EmailSet** element specified in section 2.2.3.1.14 and the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Emails" type="ArrayOfEmail" />
```

2.2.3.1.14 EmailSet Element

Type: **EmailSet** (section 2.2.3.2.15)

The **EmailSet** element is the root element for the **XML document** contained in the **PidNameExtractedEmails** property specified in section 2.2.2.3. This element contains the known entities that represent email addresses, as specified in [MS-OWEMXML].

```
<xs:element name="EmailSet" nillable="true" type="EmailSet" />
```

2.2.3.1.15 EmailString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **EmailString** element contains a string that represents an email address. It is an optional child element of the **Email** element specified in section 2.2.3.1.12.

```
<xs:element minOccurs="0" maxOccurs="1" name="EmailString" type="xs:string" />
```

2.2.3.1.16 EmailUser Element

Type: **EmailUser** (section 2.2.3.2.16)

The **EmailUser** element contains a string that represents a person with an email address. It is an optional child element of the **Attendees** element specified in section 2.2.3.1.5.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="EmailUser"
    nillable="true" type="EmailUser" />
```

2.2.3.1.17 EndTime Element

Type: **xs:dateTime** ([XMLSCHEMA2/2] section 3.2.7)

The **EndTime** element contains the date and time when the meeting ends. It is a required child element of the **Meeting** element specified in section 2.2.3.1.18.

```
<xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true" type="xs:dateTime" />
```

2.2.3.1.18 Meeting Element

Type: **Meeting** (section 2.2.3.2.17)

The **Meeting** element contains elements that represent a meeting suggestion. It is an optional child element of the **Meetings** element specified in section 2.2.3.1.19.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Meeting"
    nillable="true" type="Meeting" />
```

2.2.3.1.19 Meetings Element

Type: **ArrayOfMeeting** (section 2.2.3.2.7)

The **Meetings** element contains a list of meeting suggestions. It is an optional child element of the **MeetingSet** element specified in section 2.2.3.1.20.

```
<xs:element minOccurs="0" maxOccurs="1" name="Meetings" type="ArrayOfMeeting" />
```

2.2.3.1.20 MeetingSet Element

Type: **MeetingSet** (section 2.2.3.2.18)

The **MeetingSet** element is the root element for the **XML document** contained in the **PidNameExtractedMeetings** property specified in section 2.2.2.4. This element contains the known entities that represent meeting suggestions, as specified in [MS-OWEMXML].

```
<xs:element name="MeetingSet" nillable="true" type="MeetingSet" />
```

2.2.3.1.21 MeetingString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **MeetingString** element contains the string from the **Message object** that represents the meeting suggestion. It is an optional child element of the **Meeting** element specified in section 2.2.3.1.18.

```
<xs:element minOccurs="0" maxOccurs="1" name="MeetingString" type="xs:string" />
```

2.2.3.1.22 OriginalPhoneString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **OriginalPhoneString** element contains the original string (before normalization) from the **Message object** that represents a phone number. It is an optional child element of the **Phone** element specified in section 2.2.3.1.25.

```
<xs:element minOccurs="0" maxOccurs="1" name="OriginalPhoneString" type="xs:string" />
```

2.2.3.1.23 Person Element

Type: **Person** (section 2.2.3.2.19)

The **Person** element contains elements that represent a person's name. It is an optional child element of the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Person" type="Person" />
```

2.2.3.1.24 PersonString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **PersonString** element contains a string that represents a person's name. It is an optional child element of the **Person** element specified in section 2.2.3.1.23.

```
<xs:element minOccurs="0" maxOccurs="1" name="PersonString" type="xs:string" />
```

2.2.3.1.25 Phone Element

Type: **Phone** (section 2.2.3.2.20)

The **Phone** element contains elements that represent a phone number. It is an optional child element of the **Phones** element specified in section 2.2.3.1.26.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Phone" nillable="true" type="Phone" />
```

2.2.3.1.26 Phones Element

Type: **ArrayOfPhone** (section 2.2.3.2.8)

The **Phones** element contains a list of phone numbers. It is an optional child element of the **PhoneSet** element specified in section 2.2.3.1.27 and the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Phones" type="ArrayOfPhone" />
```

2.2.3.1.27 PhoneSet Element

Type: **PhoneSet** (section 2.2.3.2.21)

The **PhoneSet** element is the root element for the **XML document** contained in the **PidNameExtractedPhones** property specified in section 2.2.2.5. This element contains the known entities that represent phone numbers, as specified in [MS-OWEMXML].

```
<xs:element name="PhoneSet" nillable="true" type="PhoneSet" />
```

2.2.3.1.28 PhoneString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **PhoneString** element contains a string that represents a normalized phone number. The normalized phone number is based on the value of the **OriginalPhoneString** element (section 2.2.3.1.22) and is the result of modifying that value into a standard, consistent format. It is an optional child element of the **Phone** element specified in section 2.2.3.1.25.

```
<xs:element minOccurs="0" maxOccurs="1" name="PhoneString" type="xs:string" />
```

2.2.3.1.29 StartTime Element

Type: **xs:dateTime** ([XMLSCHEMA2/2] section 3.2.7)

The **StartTime** element contains the date and time when the meeting starts. It is a required child element of the **Meeting** element specified in section 2.2.3.1.18.

```
<xs:element minOccurs="1" maxOccurs="1" name="StartTime"
    nillable="true" type="xs:dateTime" />
```

2.2.3.1.30 Task Element

Type: **Task** (section 2.2.3.2.22)

The **Task** element contains elements that represent a task suggestion. It is an optional child element of the **Tasks** element specified in section 2.2.3.1.31.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Task" nillable="true" type="Task" />
```

2.2.3.1.31 Tasks Element

Type: **ArrayOfTask** (section 2.2.3.2.9)

The **Tasks** element contains a list of task suggestions. It is an optional child element of the **TaskSet** element specified in section 2.2.3.1.32.

```
<xs:element minOccurs="0" maxOccurs="1" name="Tasks" type="ArrayOfTask" />
```

2.2.3.1.32 TaskSet Element

Type: **TaskSet** (section 2.2.3.2.23)

The **TaskSet** element is the root element for the **XML document** contained in the **PidNameExtractedTasks** property specified in section 2.2.2.6. This element contains the known entities that represent task suggestions, as specified in [MS-OWEMXML].

```
<xs:element name="TaskSet" nillable="true" type="TaskSet" />
```

2.2.3.1.33 TaskString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **TaskString** element contains a string that describes the task suggestion. It is an optional child element of the **Task** element specified in section 2.2.3.1.30.

```
<xs:element minOccurs="0" maxOccurs="1" name="TaskString" type="xs:string" />
```

2.2.3.1.34 Url Element

Type: **Url** (section 2.2.3.2.24)

The **Url** element contains elements that represent a **URL**. It is an optional child element of the **Urls** element specified in section 2.2.3.1.35.

```
<xs:element minOccurs="0" maxOccurs="unbounded" name="Url" nillable="true" type="Url" />
```

2.2.3.1.35 Urls Element

Type: **ArrayOfUrl** (section 2.2.3.2.10)

The **Urls** element contains a list of **URLs**. It is an optional child element of the **UrlSet** element specified in section 2.2.3.1.36 and the **Contact** element specified in section 2.2.3.1.8.

```
<xs:element minOccurs="0" maxOccurs="1" name="Urls" type="ArrayOfUrl" />
```

2.2.3.1.36 UrlSet Element

Type: **UrlSet** (section 2.2.3.2.25)

The **UrlSet** element is the root element for the **XML document** contained in the **PidNameExtractedUrls** property specified in section 2.2.2.7. This element contains the known entities that represent **URLs**, as specified in [MS-OWEMXML].

```
<xs:element name="UrlSet" nillable="true" type="UrlSet" />
```

2.2.3.1.37 UrlString Element

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **UrlString** element contains a string that represents a **URL**. It is an optional child element of the **Url** element specified in section 2.2.3.1.34.

```
<xs:element minOccurs="0" maxOccurs="1" name="UrlString" type="xs:string" />
```

2.2.3.1.38 Version Element

Type: **Version** (section 2.2.3.3.4)

The **Version** element contains a string that describes the version of the **XML schema** that applies to the **XML document**. The value of this element MUST be "15.0.0.0". It is an optional child element of the following elements:

- **AddressSet** (section 2.2.3.1.3)

- **ContactSet** (section 2.2.3.1.10)
- **EmailSet** (section 2.2.3.1.14)
- **MeetingSet** (section 2.2.3.1.20)
- **PhoneSet** (section 2.2.3.1.27)
- **TaskSet** (section 2.2.3.1.32)
- **UrlSet** (section 2.2.3.1.36)

```
<xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
```

2.2.3.2 Complex Types

The set of common **XML schema** complex type definitions defined by this specification is summarized in the following table.

Complex type	Description
Address (section 2.2.3.2.1)	Contains information that describes an address known entity.
AddressSet (section 2.2.3.2.2)	Contains information that describes a set of address known entities.
ArrayOfAddress (section 2.2.3.2.3)	Contains a list of addresses.
ArrayOfContact (section 2.2.3.2.4)	Contains a list of contacts.
ArrayOfEmail (section 2.2.3.2.5)	Contains a list of email addresses.
ArrayOfEmailUser (section 2.2.3.2.6)	Contains a list of email users.
ArrayOfMeeting (section 2.2.3.2.7)	Contains a list of meeting suggestions.
ArrayOfPhone (section 2.2.3.2.8)	Contains a list of phone numbers.
ArrayOfTask (section 2.2.3.2.9)	Contains a list of task suggestions.
ArrayOfUrl (section 2.2.3.2.10)	Contains a list of **URLs**.
Business (section 2.2.3.2.11)	Contains information that describes a business.
Contact (section 2.2.3.2.12)	Contains information that describes a contact.
ContactSet (section 2.2.3.2.13)	Contains information that describes a set of contact known entities.
Email (section 2.2.3.2.14)	Contains information that describes an email address.
EmailSet (section 2.2.3.2.15)	Contains information that describes a set of email address known entities.
EmailUser (section 2.2.3.2.16)	Contains information that describes an email user.
Meeting (section 2.2.3.2.17)	Contains information that describes a meeting suggestion.
MeetingSet (section 2.2.3.2.18)	Contains information that describes a set of meeting suggestion known entities.

Complex type	Description
Person (section 2.2.3.2.19)	Contains information that describes a person.
Phone (section 2.2.3.2.20)	Contains information that describes a phone number.
PhoneSet (section 2.2.3.2.21)	Contains information that describes a set of phone number known entities.
Task (section 2.2.3.2.22)	Contains information that describes a task suggestion.
TaskSet (section 2.2.3.2.23)	Contains information that describes a set of task suggestion known entities.
Url (section 2.2.3.2.24)	Contains information that describes a URL.
UrlSet (section 2.2.3.2.25)	Contains information that describes a set of URL known entities.

2.2.3.2.1 Address Complex Type

The **Address** type contains information that describes a known entity that represents a postal or street address.

```
<xs:complexType name="Address">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute default="-1" name="StartIndex" type="xs:int" />
            <xs:attribute default="LatestReply" name="Position"
                type="EmailPosition" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The value of elements of this type is the string representation of a postal or street address.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

2.2.3.2.2 AddressSet Complex Type

The **AddressSet** type contains information that describes a set of known entities that represent postal or street addresses.

```
<xs:complexType name="AddressSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Addresses"
            type="ArrayOfAddress" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Addresses: An element of the **ArrayOfAddress** complex type, as defined in section 2.2.3.2.3, that contains a list of known entities that represent postal or street addresses. For more details, see section 2.2.3.1.2.

2.2.3.2.3 ArrayOfAddress Complex Type

The **ArrayOfAddress** type contains zero or more **Address** elements.

```
<xs:complexType name="ArrayOfAddress">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Address"
            nillable="true" type="Address" />
    </xs:sequence>
</xs:complexType>
```

Address: An element of the **Address** complex type, as defined in section 2.2.3.2.1, that contains information about a single postal or street address. For more details, see section 2.2.3.1.1.

2.2.3.2.4 ArrayOfContact Complex Type

The **ArrayOfContact** type contains zero or more **Contact** elements.

```
<xs:complexType name="ArrayOfContact">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Contact"
            nillable="true" type="Contact" />
    </xs:sequence>
</xs:complexType>
```

Contact: An element of the **Contact** complex type, as defined in section 2.2.3.2.12, that contains information about a single contact. For more details, see section 2.2.3.1.8.

2.2.3.2.5 ArrayOfEmail Complex Type

The **ArrayOfEmail** type contains zero or more **Email** elements.

```
<xs:complexType name="ArrayOfEmail">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Email"
            nillable="true" type="Email" />
    </xs:sequence>
</xs:complexType>
```

Email: An element of the **Email** complex type, as defined in section 2.2.3.2.14, that contains information about a single email address. For more details, see section 2.2.3.1.12.

2.2.3.2.6 ArrayOfEmailUser Complex Type

The **ArrayOfEmailUser** type contains zero or more **EmailUser** elements.

```
<xs:complexType name="ArrayOfEmailUser">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="EmailUser"
```

```
            nillable="true" type="EmailUser" />
    </xs:sequence>
</xs:complexType>
```

EmailUser: An element of the **EmailUser** complex type, as defined in section 2.2.3.2.16, that contains information about a single email user. For more details, see section 2.2.3.1.16.

2.2.3.2.7 ArrayOfMeeting Complex Type

The **ArrayOfMeeting** type contains zero or more **Meeting** elements.

```
<xs:complexType name="ArrayOfMeeting">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Meeting"
            nillable="true" type="Meeting" />
    </xs:sequence>
</xs:complexType>
```

Meeting: An element of the **Meeting** complex type, as defined in section 2.2.3.2.17, that contains information about a single meeting. For more details, see section 2.2.3.1.18.

2.2.3.2.8 ArrayOfPhone Complex Type

The **ArrayOfPhone** type contains zero or more **Phone** elements.

```
<xs:complexType name="ArrayOfPhone">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Phone"
            nillable="true" type="Phone" />
    </xs:sequence>
</xs:complexType>
```

Phone: An element of the **Phone** complex type, as defined in section 2.2.3.2.20, that contains information about a single phone number. For more details, see section 2.2.3.1.25.

2.2.3.2.9 ArrayOfTask Complex Type

The **ArrayOfTask** type contains zero or more **Task** elements, as specified in section 2.2.3.1.30.

```
<xs:complexType name="ArrayOfTask">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Task"
            nillable="true" type="Task" />
    </xs:sequence>
</xs:complexType>
```

Task: An element of the **Task** complex type, as defined in section 2.2.3.2.22, that contains information about a single task suggestion.

2.2.3.2.10 ArrayOfUrl Complex Type

The **ArrayOfUrl** type contains zero or more **Url** elements, as specified in section 2.2.3.1.34.

```
<xs:complexType name="ArrayOfUrl">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Url"
```

```
            nillable="true" type="Url" />
    </xs:sequence>
</xs:complexType>
```

Url: An element of the **Url** complex type, as defined in section 2.2.3.2.24, that contains information about a single **URL**.

2.2.3.2.11 Business Complex Type

The **Business** type contains information that describes a known entity that represents a business associated with a contact.

```
<xs:complexType name="Business">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="BusinessString"
            type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
```

BusinessString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains the name of the business. For more details, see section 2.2.3.1.7.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

2.2.3.2.12 Contact Complex Type

The **Contact** type contains information that describes a known entity that represents a contact.

```
<xs:complexType name="Contact">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Person" type="Person" />
        <xs:element minOccurs="0" maxOccurs="1" name="Business" type="Business" />
        <xs:element minOccurs="0" maxOccurs="1" name="Phones"
            type="ArrayOfPhone" />
        <xs:element minOccurs="0" maxOccurs="1" name="Urls" type="ArrayOfUrl" />
        <xs:element minOccurs="0" maxOccurs="1" name="Emails"
            type="ArrayOfEmail" />
        <xs:element minOccurs="0" maxOccurs="1" name="Addresses"
            type="ArrayOfAddress" />
        <xs:element minOccurs="0" maxOccurs="1" name="ContactString"
            type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Person: An element of the **Person** complex type, as defined in section 2.2.3.2.19, that, if present, contains information about the person represented by the contact. For more details, see section 2.2.3.1.23.

Business: An element of the **Business** complex type, as defined in section 2.2.3.2.11, that, if present, contains the name of a business associated with the contact. For more details, see section 2.2.3.1.6.

Phones: An element of the **ArrayOfPhone** complex type, as defined in section 2.2.3.2.8, that, if present, contains the phone numbers associated with the contact. For more details, see section 2.2.3.1.26.

Urls: An element of the **ArrayOfUrl** complex type, as defined in section 2.2.3.2.10, that, if present, contains the **URLs** associated with the contact. For more details, see section 2.2.3.1.35.

Emails: An element of the **ArrayOfEmail** complex type, as defined in section 2.2.3.2.5, that, if present, contains the email addresses associated with the contact. For more details, see section 2.2.3.1.13.

Addresses: An element of the **ArrayOfAddress** complex type, as defined in section 2.2.3.2.3, that, if present, contains the postal or street addresses associated with the contact. For more details, see section 2.2.3.1.2.

ContactString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that, if present, contains the text from the **Message object** that indicates a contact. For more details, see section 2.2.3.1.11.

2.2.3.2.13 ContactSet Complex Type

The **ContactSet** type contains information that describes a set of known entities that represent contacts.

```
<xs:complexType name="ContactSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Contacts"
            type="ArrayOfContact" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Contacts: An element of the **ArrayOfContact** complex type, as defined in section 2.2.3.2.4, that contains a list of known entities that represent contacts. For more details, see section 2.2.3.1.9.

2.2.3.2.14 Email Complex Type

The **Email** type contains information that describes a known entity that represents an email address.

```
<xs:complexType name="Email">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="EmailString"
            type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
```

EmailString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains an email address. For more details, see section 2.2.3.1.15.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

2.2.3.2.15 EmailSet Complex Type

The **EmailSet** type contains information that describes a set of known entities that represent email addresses.

```
<xs:complexType name="EmailSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Emails"
            type="ArrayOfEmail" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Emails: An element of the **ArrayOfEmail** complex type, as defined in section 2.2.3.2.5, that contains a list of known entities that represent email addresses. For more details, see section 2.2.3.1.13.

2.2.3.2.16 EmailUser Complex Type

The **EmailUser** type contains information that describes an email user.

```
<xs:complexType name="EmailUser">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="Id" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The value of an element of this type is the name of the user.

Id: An attribute of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains a unique identifier for the user, such as the user's **Simple Mail Transfer Protocol (SMTP)** address. For more details, see section 2.2.3.4.1.

2.2.3.2.17 Meeting Complex Type

The **Meeting** type contains information that describes a known entity that represents a meeting suggestion.

```
<xs:complexType name="Meeting">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="MeetingString"
            type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Attendees"
            type="ArrayOfEmailUser" />
        <xs:element minOccurs="1" maxOccurs="1" name="StartTime" nillable="true"
            type="xs:dateTime" />
        <xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true"
            type="xs:dateTime" />
    </xs:sequence>
    <xs:attribute name="Location" type="xs:string" />
    <xs:attribute name="Subject" type="xs:string" />
```

```
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
```

MeetingString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains the text from the **Message object** that indicates a meeting suggestion. For more details, see section 2.2.3.1.21.

Attendees: An element of the **ArrayOfEmailUser** complex type, as defined in section 2.2.3.2.6, that contains the attendees of the meeting. For more details, see section 2.2.3.1.5.

StartTime: An element of type **xs:dateTime**, as defined in ([XMLSCHEMA2/2] section 3.2.7, that contains the date and time the meeting is scheduled to start. For more details, see section 2.2.3.1.29.

EndTime: An element of type **xs:dateTime** that contains the date and time the meeting is scheduled to end. For more details, see section 2.2.3.1.17.

Location: An attribute of type **xs:string** that contains the location of the meeting. For more details, see section 2.2.3.4.2.

Subject: An attribute of type **xs:string** that contains the subject of the meeting. For more details, see section 2.2.3.4.5.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the Message object. For more details, see section 2.2.3.4.3.

2.2.3.2.18 MeetingSet Complex Type

The **MeetingSet** type contains information that describes a set of known entities that represent meeting suggestions.

```
<xs:complexType name="MeetingSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Meetings"
            type="ArrayOfMeeting" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Meetings: An element of the **ArrayOfMeeting** complex type, as defined in section 2.2.3.2.7, that contains a list of known entities that represent meetings. For more details, see section 2.2.3.1.19.

2.2.3.2.19 Person Complex Type

The **Person** type contains information that describes a known entity that represents a person associated with a contact.

```
<xs:complexType name="Person">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="PersonString"
```

```
            type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
```

PersonString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains the name of the person. For more details, see section 2.2.3.1.24.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

2.2.3.2.20 Phone Complex Type

The **Phone** type contains information that describes a known entity that represents a phone number.

```
<xs:complexType name="Phone">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="PhoneString"
            type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="OriginalPhoneString"
            type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
    <xs:attribute default="Unspecified" name="Type" type="PhoneType" />
</xs:complexType>
```

PhoneString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains the normalized phone number associated with the known entity. For more details, see section 2.2.3.1.28.

OriginalPhoneString: An element of type **xs:string** that contains the original string from the **Message object** that represents the phone number associated with the known entity. For more details, see section 2.2.3.1.22.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the Message object. For more details, see section 2.2.3.4.3.

Type: An attribute of the **PhoneType** simple type, as defined in section 2.2.3.3.2, that indicates the type of phone number. For more details, see section 2.2.3.4.6.1.

2.2.3.2.21 PhoneSet Complex Type

The **PhoneSet** type contains information that describes a set of known entities that represent phone numbers.

```
<xs:complexType name="PhoneSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
```

```
        <xs:element minOccurs="0" maxOccurs="1" name="Phones"
            type="ArrayOfPhone" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Phones: An element of the **ArrayOfPhone** complex type, as defined in section 2.2.3.2.8, that contains a list of known entities that represent phone numbers. For more details, see section 2.2.3.1.26.

2.2.3.2.22 Task Complex Type

The **Task** type contains information that describes a known entity that represents a task suggestion.

```
<xs:complexType name="Task">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="TaskString"
            type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Assignees"
            type="ArrayOfEmailUser" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
```

TaskString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains a string that describes the task suggestion. For more details, see section 2.2.3.1.33.

Assignees: An element of the **ArrayOfEmailUser** complex type, as defined in section 2.2.3.2.6, that contains a list of assignees for the task. For more details, see section 2.2.3.1.4.

StartIndex: An attribute of type **xs:int**, as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

2.2.3.2.23 TaskSet Complex Type

The **TaskSet** type contains information that describes a set of known entities that represent task suggestions.

```
<xs:complexType name="TaskSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Tasks" type="ArrayOfTask" />
  </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Tasks: An element of the **ArrayOfTask** complex type, as defined in section 2.2.3.2.9, that contains a list of known entities that represent tasks. For more details, see section 2.2.3.1.31.

2.2.3.2.24 Url Complex Type

The **Url** type contains information that describes a known entity that represents a **URL**.

```
<xs:complexType name="Url">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="UrlString"
            type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
    <xs:attribute default="Unspecified" name="Type" type="UrlType" />
</xs:complexType>
```

UrlString: An element of type **xs:string**, as defined in [XMLSCHEMA2/2] section 3.2.1, that contains the URL. For more details, see section 2.2.3.1.37.

StartIndex: An attribute of type **xs:int,** as defined in [XMLSCHEMA2/2] section 3.3.17, that indicates the location of the entity, relative to the value of the **Position** attribute. For more details, see section 2.2.3.4.4.

Position: An attribute of the **EmailPosition** simple type, as defined in section 2.2.3.3.1, that indicates the location of the entity within the **Message object**. For more details, see section 2.2.3.4.3.

Type: An attribute of the **UrlType** simple type, as defined in section 2.2.3.3.3, that indicates the type of URL. For more details, see section 2.2.3.4.6.2.

2.2.3.2.25 UrlSet Complex Type

The **UrlSet** type contains information that describes a set of known entities that represent **URLs**.

```
<xs:complexType name="UrlSet">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Version" type="Version" />
        <xs:element minOccurs="0" maxOccurs="1" name="Urls" type="ArrayOfUrl" />
    </xs:sequence>
</xs:complexType>
```

Version: An element of the **Version** simple type, as defined in section 2.2.3.3.4, that indicates the version of the **XML schema** that applies to the parent **XML document**. For more details, see section 2.2.3.1.38.

Urls: An element of the **ArrayOfUrl** complex type, as defined in section 2.2.3.2.10, that contains a list of known entities that represent URLs. For more details, see section 2.2.3.1.35.

2.2.3.3 Simple Types

The set of common **XML schema** simple type definitions defined by this specification is summarized in the following table.

Simple type	Description
EmailPosition (section 2.2.3.3.1)	Specifies the portion of a **Message object** where a known entity is located.

Simple type	Description
PhoneType (section 2.2.3.3.2)	Specifies the type of a phone number.
UrlType (section 2.2.3.3.3)	Specifies the type of a **URL**.
Version (section 2.2.3.3.4)	Contains information that describes the version of the XML schema that applies to the **XML document**.

2.2.3.3.1 EmailPosition Simple Type

The **EmailPosition** type is used by the **Position** attribute, as specified in section 2.2.3.4.3, to indicate the portion of the **Message object** where a known entity is located.

```
<xs:simpleType name="EmailPosition">
    <xs:restriction base="xs:string">
        <xs:enumeration value="LatestReply" />
        <xs:enumeration value="Subject" />
        <xs:enumeration value="Signature" />
        <xs:enumeration value="Other" />
    </xs:restriction>
</xs:simpleType>
```

The possible values for attributes of this type are specified in the following table.

Value	Meaning
LatestReply	The known entity is located in the most recent reply in a **conversation**.
Subject	The known entity is located in the subject of the message.
Signature	The known entity is located in the sender's signature within the most recent reply in a conversation.
Other	The known entity is located in the body of the message and does not meet the criteria for the other three values.

2.2.3.3.2 PhoneType Simple Type

The **PhoneType** type is used by the **Type** attribute, as specified in section 2.2.3.4.6.1, on the **Phone** element, as specified in section 2.2.3.1.25, to indicate the type of phone number associated with the known entity.

```
<xs:simpleType name="PhoneType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Unspecified" />
        <xs:enumeration value="Home" />
        <xs:enumeration value="Mobile" />
        <xs:enumeration value="Work" />
        <xs:enumeration value="Fax" />
    </xs:restriction>
</xs:simpleType>
```

The possible values for attributes of this type are specified in the following table.

Value	Meaning
Unspecified	The type of phone number is unavailable.
Home	The phone number is for a home phone.
Mobile	The phone number is for a mobile phone.
Work	The phone number is for a work phone.
Fax	The phone number is for a fax machine.

2.2.3.3.3 UrlType Simple Type

The **UrlType** type is used by the **Type** attribute, as specified in section 2.2.3.4.6.2, on the **Url** element, as specified in section 2.2.3.1.34, to indicate the type of **URL** associated with the known entity.

```
<xs:simpleType name="UrlType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Unspecified" />
        <xs:enumeration value="Url" />
        <xs:enumeration value="Filename" />
    </xs:restriction>
</xs:simpleType>
```

The possible values for attributes of this type are specified in the following table.

Value	Meaning
Unspecified	The type of URL is unavailable.
Url	The URL is for a remote resource, such as a website or a file on a network share.
Filename	The URL is for a local file.

2.2.3.3.4 Version Simple Type

The **Version** type contains information about the version of the **XML schema** that applies to the parent **XML document**.

```
<xs:simpleType name="Version">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]+\.[0-9]+(\.[0-9]+(\.[0-9]+)?)?"/>
  </xs:restriction>
</xs:simpleType>
```

2.2.3.4 Attributes

The set of common **XML schema** attribute definitions defined by this specification is summarized in the following table.

Attribute	Description
Id (section 2.2.3.4.1)	Specifies a unique identifier for an email user.
Location (section 2.2.3.4.2)	Specifies the location of a meeting.
Position (section 2.2.3.4.3)	Specifies the location of a known entity within a **Message object**.
StartIndex (section 2.2.3.4.4)	Specifies the location of a known entity, relative to the value of the **Position** attribute.
Subject (section 2.2.3.4.5)	Specifies the subject of a meeting.
Type (section 2.2.3.4.6)	Specifies the type of a phone number or **URL**.

2.2.3.4.1 Id Attribute

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **Id** attribute specifies a unique identifier for an email user, such as an **SMTP** address. It is used on elements of the **EmailUser** type, as specified in section 2.2.3.2.16.

```
<xs:attribute name="Id" type="xs:string" />
```

2.2.3.4.2 Location Attribute

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **Location** attribute specifies the location of a meeting. It is used on elements of the **Meeting** type, as specified in section 2.2.3.2.17.

```
<xs:attribute name="Location" type="xs:string" />
```

2.2.3.4.3 Position Attribute

Type: **EmailPosition** (section 2.2.3.3.1)

The **Position** attribute specifies the location of a known entity within a **Message object**. It is used by the following complex types:

- **Address**, as specified in section 2.2.3.2.1
- **Business**, as specified in section 2.2.3.2.11
- **Email**, as specified in section 2.2.3.2.14
- **Meeting**, as specified in section 2.2.3.2.17
- **Person**, as specified in section 2.2.3.2.19
- **Phone**, as specified in section 2.2.3.2.20
- **Task**, as specified in section 2.2.3.2.22
- **Url**, as specified in section 2.2.3.2.24

```
<xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
```

2.2.3.4.4 StartIndex Attribute

Type: **xs:int** ([XMLSCHEMA2/2] section 3.3.17)

The **StartIndex** attribute is an integer that specifies the location of a known entity, relative to the value of the **Position** attribute, as specified in section 2.2.3.4.3. It is used by the following complex types:

- **Address**, as specified in section 2.2.3.2.1
- **Business**, as specified in section 2.2.3.2.11
- **Email**, as specified in section 2.2.3.2.14
- **Meeting**, as specified in section 2.2.3.2.17
- **Person**, as specified in section 2.2.3.2.19
- **Phone**, as specified in section 2.2.3.2.20
- **Task**, as specified in section 2.2.3.2.22
- **Url**, as specified in section 2.2.3.2.24

The meaning of this attribute depends on the value of the **Position** attribute on the containing element.

If the **Position** attribute has a value of "Subject", the value of the **StartIndex** attribute is the number of characters from the beginning of the subject of the message.

For all other values of the **Position** attribute, the value of the **StartIndex** attribute is the number of characters from the beginning of the plain text representation of the body of the message.

A value of -1 indicates that the exact location of the known entity is unavailable.

```
<xs:attribute default="-1" name="StartIndex" type="xs:int" />
```

2.2.3.4.5 Subject Attribute

Type: **xs:string** ([XMLSCHEMA2/2] section 3.2.1)

The **Subject** attribute specifies the subject of a meeting. It is used by elements of the **Meeting** type, as specified in section 2.2.3.2.17.

```
<xs:attribute name="Subject" type="xs:string" />
```

2.2.3.4.6 Type Attribute

The type and meaning of the **Type** attribute depend on the type of the element that contains the attribute. The **Type** attribute is used on elements of the following types:

- **Phone**, as specified in section 2.2.3.2.20
- **Url**, as specified in section 2.2.3.2.24

For details on the **Type** attribute when used on an element of type **Phone**, see section 2.2.3.4.6.1.
For details on the **Type** attribute when used on an element of type **Url**, see section 2.2.3.4.6.2.

2.2.3.4.6.1 Type Attribute (Phone)

Type: **PhoneType** (section 2.2.3.3.2)

The **Type** attribute specifies the type of phone number associated with a known entity. It is used on elements of the **Phone** type, as specified in section 2.2.3.2.20.

```
<xs:attribute default="Unspecified" name="Type" type="PhoneType" />
```

2.2.3.4.6.2 Type Attribute (Url)

Type: **UrlType** (section 2.2.3.3.3)

The **Type** attribute specifies the type of **URL** associated with a known entity. It is used on elements of the **Url** type, as specified in section 2.2.3.2.24.

```
<xs:attribute default="Unspecified" name="Type" type="UrlType" />
```

2.2.4 Mail App Configuration Data

Mail app–specific configuration data is stored as **dictionary** configuration data in the user's inbox, as specified in [MS-OXOCFG] section 2.2.5.1. The value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) on the **Message object** that stores the dictionary configuration data is set to "IPM.Configuration.ClientExtension.<ID>", where "<ID>" is replaced with a value derived from the value of the **Id** child element of the **OfficeApp** element that represents the **mail add-in** in the mail add-in manifest, as specified in [MS-OWEMXML]. The value is derived by using the following procedure:

1. Replace any uppercase alphabetic characters in the value of the **Id** element with their lowercase equivalent. For example, 'A' becomes 'a'.
2. Remove any braces ('{' or '}') from the value.
3. Remove any dashes ('-') from the value.
4. If present, remove the "urn:uuid" prefix from the value.

The client-specific configuration data is stored in **JavaScript Object Notation (JSON)** in a name-value pair in an **e** element in the **PidTagRoamingDictionary** property ([MS-OXOCFG] section 2.2.2.2), as specified in [MS-OXOCFG] section 2.2.5.1. The name of the name-value pair is ExtensionSettings, and the value is the JSON object that represents the mail app's settings. Note that the name-value pairs within the JSON object are specific to the mail add-in.

2.2.5 Mail App Custom Properties

Mail app–specific custom properties on a **Message object** are stored as a **JSON** object in a string **named property** on the Message object. The property is defined as follows.

Property set: PS_PUBLIC_STRINGS {00020329-0000-0000-C000-000000000046}

Property name: cecp-<ID>

Data type: **PtypString**, 0x001F ([MS-OXCDATA] section 2.11.1)

The <ID> portion of the **property name** is replaced by a value derived from the value of the **Id** child element of the **OfficeApp** element that represents the **mail add-in** in the mail add-in manifest, as specified in [MS-OWEMXML]. The value is derived by using the following procedure:

1. Replace any uppercase alphabetic characters in the value of the **Id** element with their lowercase equivalent. For example, 'A' becomes 'a'.
2. Remove any braces ('{' or '}') from the value.
3. If present, remove the "urn:uuid" prefix from the value.

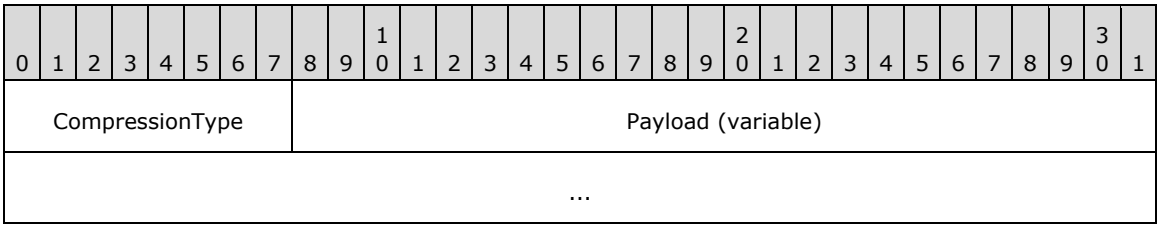
The value of the property is a JSON object that contains name-value pairs, where the name is the name of the custom property and the value is the value of the custom property. The value is limited to a maximum of 2,500 characters.

2.2.6 Derived Web Services Identifier

The derived web services identifier is used to generate an item identifier as specified in [MS-OXWSCORE] section 2.2.4.25. The format of a derived web services identifier is a hexadecimal string representation of the **DerivedWSId** structure specified in section 2.2.6.1, encoded with **base64 encoding**.

2.2.6.1 DerivedWSId Structure

The **DerivedWSId** structure is used to derive a web services item identifier.

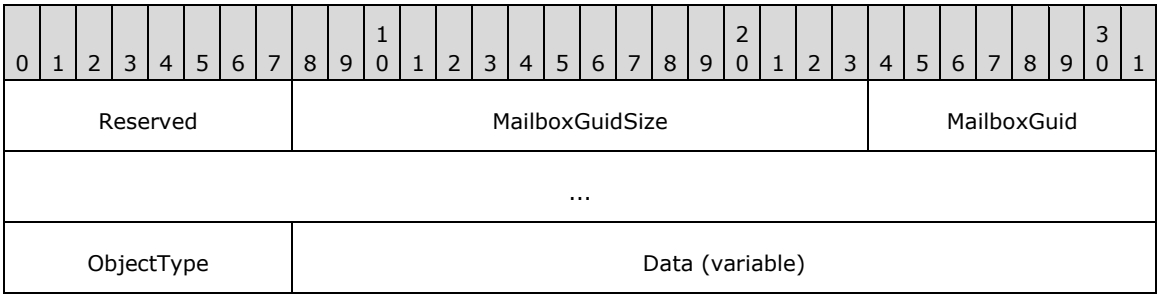


CompressionType (1 byte): A byte that specifies the compression status of the **Payload** field. This field is set to 0x00 if the **Payload** field is not compressed. It is set to 0x01 if the **Payload** field is compressed.

Payload (variable): An array of bytes that contains a structure. If the value of the **CompressionType** field is 0x00, this field contains a **DerivedId** structure. If the value of the **CompressionType** field is 0x01, this field contains a **DerivedId** structure compressed with the compression algorithm specified in section 3.1.4.5.

2.2.6.1.1 DerivedId Structure

The **DerivedId** structure is used to generate the **Payload** field of the **DerivedWSId** structure (section 2.2.6.1).



...
AttachmentData (variable, optional)

Reserved (1 byte): A byte that is set to 0x03 if the **Data** field contains **ItemData** structure specified in section 2.2.6.1.1.1 or **RecurrenceItemData** structure specified in section 2.2.6.1.1.2; otherwise, set to 0x04 if the **Data** field contains **ConversationData** structure specified in section 2.2.6.1.1.3.

MailboxGuidSize (2 bytes): A 16-bit integer that specifies the number of bytes in the **MailboxGuid** field. This field is written in **little-endian** order.

MailboxGuid (variable): A string that contains the **mailbox GUID** in **GUIDString** format.

ObjectType (1 byte): A byte that is set to 0x01 if the derived web services identifier is for a single occurrence of a **Recurring Calendar object**; otherwise, set to 0x00.

Data (variable): An array of bytes that contains a structure. This field contains a **RecurrenceItemData** structure specified in section 2.2.6.1.1.2 if the derived web services identifier is for a single occurrence of a Recurring Calendar object, or an **ItemData** structure, as specified in section 2.2.6.1.1.1; otherwise, it contains an **ConversationData** structure specified in section 2.2.6.1.1.3 if the derived web services identifier is for a Conversation object.

AttachmentData (variable, optional): An array of bytes that contains a structure. This field contains an **AttachmentData** structure specified in section 2.2.6.1.1.4, if the identifier represents an attachment of the item rather than the item itself.

2.2.6.1.1.1 ItemData Structure

The **ItemData** structure is used in the **Data** field of the **DerivedId** structure, as specified in section 2.2.6.1.1, when the derived web services identifier being generated is for an **Email object**, a **Calendar object** that is not recurring, or a **Recurring Calendar object**.

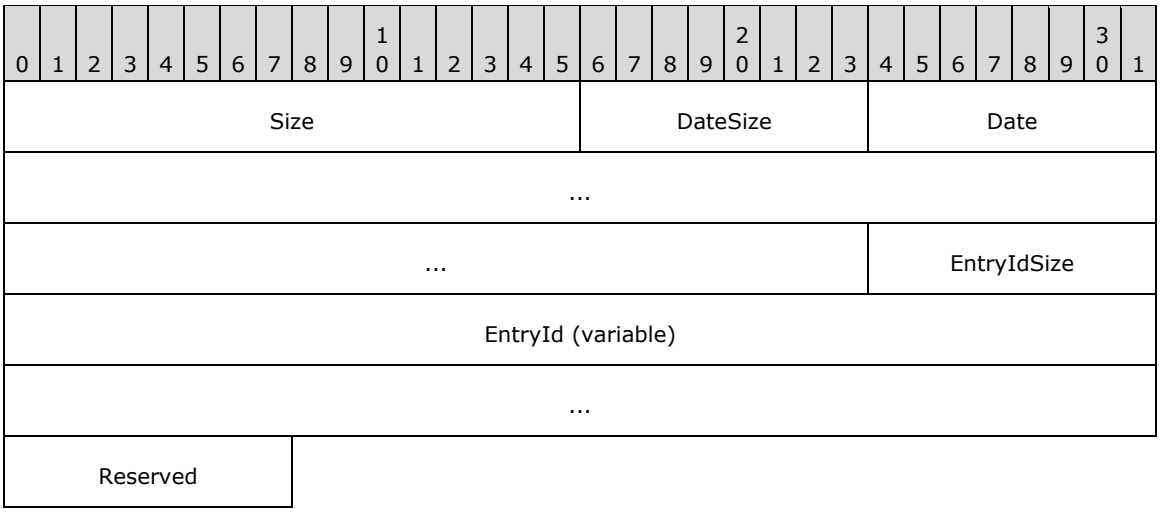
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
EntryIdSize																EntryId (variable)															
...																															

EntryIdSize (2 bytes): An integer that is set to the number of bytes in the **EntryId** field. This field is written in **little-endian** order.

EntryId (variable): An array of bytes that contains the value of the **Message EntryID** structure for the item, as specified in [MS-OXCDATA] section 2.2.4.2.

2.2.6.1.1.2 RecurrenceItemData Structure

The **RecurrenceItemData** structure is used in the **Data** field of the **DerivedId** structure, as specified in section 2.2.6.1.1, when the derived web services identifier being generated is for a single occurrence of a **Recurring Calendar object**.



Size (2 bytes): An integer that is set to the size of the **DateSize**, **Date**, **EntryIdSize**, **EntryId**, and **Reserved** fields in the **RecurrenceItemData** structure. This field is written in **little-endian** order.

DateSize (1 byte): A byte that is set to the size of the **Date** field, which is 0x08.

Date (8 bytes): A **ULONGLONG** ([MS-DTYP]) that specifies the number of 100-nanosecond intervals between 12:00:00 midnight, January 1, 0001, to 12:00:00 midnight on the date of the occurrence. This field is written in **big-endian** order.

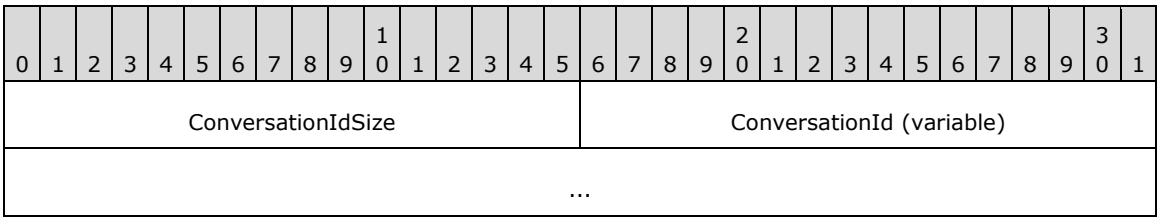
EntryIdSize (1 byte): A byte that is set to the size of the **EntryId** field.

EntryId (variable): An array of bytes that contains the value of the **Message EntryID** structure for the item, as specified in [MS-OXCDATA] section 2.2.4.2.

Reserved (1 byte): A byte that MUST be set to 0x10.

2.2.6.1.1.3 ConversationData Structure

The **ConversationData** structure is used in the **Data** field of the **DerivedId** structure, as specified in section 2.2.6.1.1, when the derived web services identifier being generated is for a Conversation object.

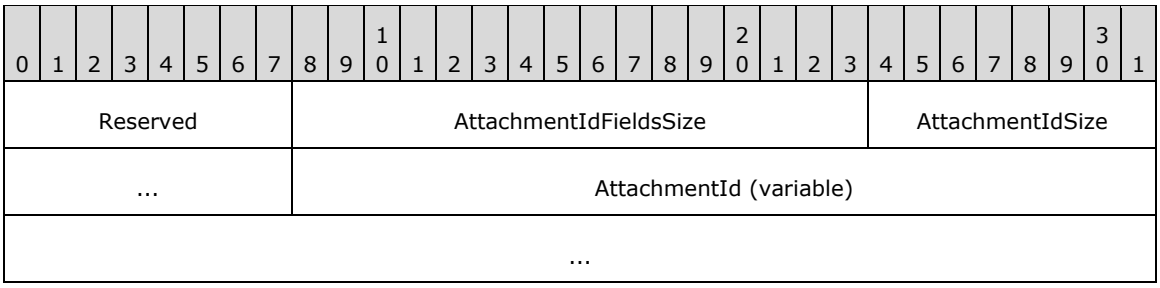


ConversationIdSize (2 bytes): An integer that is set to the number of bytes in the **ConversationId** field. This field is written in **little-endian** order.

ConversationId (variable): An array of bytes that contains the value of the **PidTagConversationId** property ([MS-OXOMSG] section 2.2.1.2) for the conversation.

2.2.6.1.1.4 AttachmentData Structure

The **AttachmentData** structure is used in the **AttachmentData** field of the **DerivedId** structure, as specified in section 2.2.6.1.1.



Reserved (1 byte): A byte that MUST be set to 0x01.

AttachmentIdFieldsSize (2 bytes): An integer that is set to the sum of 2 and the size of the **AttachmentId** field.

AttachmentIdSize (2 bytes): An integer that is set to the size of the **AttachmentId** field.

AttachmentId (variable): An array of bytes that contains **PidTagRecordKey** property ([MS-OXCPRPT] section 2.2.1.8).

3 Protocol Details

3.1 Client Details

The client uses this protocol to request notification of server-side changes to the list of enabled mail add-ins in the user's **mailbox**, to retrieve data from the server to use in evaluating which mail add-ins are applicable to the displayed message, and to retrieve data that is requested by mail add-ins.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

The client SHOULD create a table view, as specified in [MS-OXCTABL] section 3.1.4.1, for the **FAI contents table** in the user's inbox that is restricted to items that have the value "IPM.Configuration.ExtensionMasterTable" in the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3). The client SHOULD subscribe to a **TableModified** event notification for the table, as specified in [MS-OXCNOTIF] section 3.2.4.2.

3.1.4 Higher-Layer Triggered Events

The client uses this protocol to respond to the events specified in the following table.

Event	Details
The client displays a message.	Section 3.1.4.1
A **mail add-in** accesses configuration data.	Section 3.1.4.2
A mail add-in accesses custom properties on the current message.	Section 3.1.4.3
A mail add-in requests known entities on the current message.	Section 3.1.4.4
A mail add-in requests the web services identifier of the current message.	Section 3.1.4.5

Mail add-ins request data via the JavaScript API for Office, as described in [MSDN-JavaScriptApiOffice].

3.1.4.1 Client Displays a Message

When a client displays an **Email object** or a **Calendar object**, it SHOULD check the values of the known entity properties specified in section 2.2.2.1 through section 2.2.2.7 on that object for the presence of known entities. If a property is not present on the object, the client treats that object as having no known entities of the corresponding type.

The presence or absence of known entities is used in the evaluation of rules that use the **ItemHasKnownEntity** complex type for displaying mail add-ins, as specified in [MS-OWEMXML]. The

relationship of the properties and the types of known entities are specified in the following table. The known entity types are specified in [MS-OWEMXML].

Property	Known entity type
PidNameExtractedAddresses (section 2.2.2.1)	Address
PidNameExtractedContacts (section 2.2.2.2)	Contact
PidNameExtractedEmails (section 2.2.2.3)	EmailAddress
PidNameExtractedMeetings (section 2.2.2.4)	MeetingSuggestion
PidNameExtractedPhones (section 2.2.2.5)	PhoneNumber
PidNameExtractedTasks (section 2.2.2.6)	TaskSuggestion
PidNameExtractedUrls (section 2.2.2.7)	Url

3.1.4.2 Mail App Accesses Configuration Data

When a **mail add-in** accesses its configuration data, the client performs the following steps:

1. Looks up the identifier for the mail add-in in the manifest, as specified in [MS-OWEMXML].
2. Generates a value for the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) by using the procedure specified in section 2.2.4.
3. Opens the configuration data message with the value generated in step 2 in the **PidTagMessageClass** property by using the procedure specified in [MS-OXOCFG] section 3.1.4.1.
4. Reads the configuration data from the **PidTagRoamingDictionary** property ([MS-OXOCFG] section 2.2.2.2) on the configuration data message opened in step 3 by using the procedure specified in [MS-OXOCFG] section 3.1.4.1.1.
5. Returns the configuration data to the mail add-in.

If a mail add-in changes any of its configuration data, the client updates the **JSON** object, as specified in section 2.2.4, with the new information and updates the **dictionary** on the configuration data message as specified in [MS-OXOCFG] section 3.1.4.2.1. The client saves the changes to the configuration data message as specified in [MS-OXOCFG] section 3.1.4.2.

3.1.4.3 Mail App Accesses Custom Properties

When a **mail add-in** accesses its custom properties on a **Message object** or **Calendar object**, the client performs the following steps:

1. Looks up the identifier for the mail add-in in the manifest, as specified in [MS-OWEMXML].
2. Generates a **property name** as specified in section 2.2.5.
3. Uses the procedures specified in [MS-OXCPRPT] section 3.1.4.1 and [MS-OXCPRPT] section 3.1.4.5 to retrieve the value of the **named property**, which contains a **JSON** object that contains the mail app's custom properties.
4. Returns the custom properties to the mail add-in.

If a mail add-in changes any of its custom properties, the client updates the JSON object, as specified in section 2.2.5, with the new information and writes it into the named property using the procedure specified in [MS-OXCPRPT] section 3.1.4.3.

3.1.4.4 Mail App Requests Known Entities

When a **mail add-in** requests known entities on a **Message object** or **Calendar object**, the client retrieves the **XML document** from the properties specified in section 2.2.2. The client then checks the value of the **Version** element (section 2.2.3.1.38). If the value is not 15.0.0.0, the client SHOULD ignore any known entities contained within the XML document. Otherwise, the client returns the known entities contained within the XML document to the mail add-in. If the mail add-in requests MeetingSuggestion known entities, the values of the **StartTime** (section 2.2.3.1.29) and **EndTime** (section 2.2.3.1.17) elements are interpreted according to the procedure specified in section 3.1.4.4.1 before returning the known entity to the mail add-in. The relationship of the properties and the types of known entities is specified in the table in section 3.1.4.1.

3.1.4.4.1 Interpreting xs:dateTime Type Values in MeetingSuggestion Entities

Clients SHOULD examine the values of the **StartTime** (section 2.2.3.1.29) and **EndTime** (section 2.2.3.1.17) elements to determine whether they are encoded dates. Encoded dates are used to represent incomplete dates. There are two types of encoded dates: precise dates and relative dates.

Precise dates are dates that are missing one or more of either day, month, or year, such as May 25 or June 2012. Relative dates are dates that defined relative to when a **Message object** was sent, such as "tomorrow", "next Friday", or "in two weeks".

If the date portion of the value is greater than June 15, 1436, the date is not an encoded date and requires no interpretation. The value in the element is returned to the **mail add-in**. If the date portion of the value is less than or equal to June 15, 1436, the date is encoded and MUST be interpreted according to the following procedure before returning it to the mail add-in.

1. The client determines the number of days from January 1, 0001 to the date portion of the value. That number is evaluated as an integer.

2. The integer value from step 1 is represented in binary format, and all but the 18 least significant bits are ignored to generate an 18-bit integer.

3. The 3 most significant bits are examined. If they are set to 000, the encoded date is a precise date. The 15 least significant bits are interpreted as specified in section 3.1.4.4.1.1. If they are set to 001, the encoded date is a relative date. The 15 least significant bits are interpreted as specified in section 3.1.4.4.1.2.

3.1.4.4.1.1 Interpreting Precise Dates

Clients interpret the 15 least significant bits of the integer value generated in step 2 of the procedure specified in section 3.1.4.4.1 as a precise date according to the following procedure.

1. The value is split into two values. The 3 most significant bits represent the type, and the 12 least significant bits represent the value.

2. If the type bits are set to 110, the value bits represent a month and year. The 7 most significant bits of the value are converted to a decimal value, which represents the two-digit year. The next 4 most significant bits of the value are converted to a decimal value, which represents the month. The least significant bit is ignored.

3. If the type bits are set to 100, the value bits represent a year. The 7 most significant bits of the value are converted to a decimal value, which represents the two-digit year. The 5 least significant bits are ignored.

4. If the type bits are set to 011, the value bits represent a month and day. The 4 most significant bits of the value are converted to a decimal value, which represents the month. The next 5 most significant bits of the value are converted to a decimal value, which represents the day of the month. The 3 least significant bits are ignored.

5. If the type bits are set to 010, the value bits represent a month. The 4 most significant bits of the value are converted to a decimal value, which represents the month. The 8 least significant bits are ignored.

6. If the type bits are set to 001, the value bits represent a day. The 4 most significant bits of the value are ignored. The next 5 most significant bits of the value are converted to a decimal value, which represents the day of the month. The 3 least significant bits are ignored.

Year values are encoded as two-digit values. Because of this, values greater than or equal to 50 are added to 1900 to determine the year. Values less than 50 are added to 2000.

Finally, the client SHOULD fill in the missing portions of the date so that the date reflects the next available date and time that matches the specified portions of the date based on the sent date and time of the **Message object**. For example, if the interpreted value is "January 23" and the Message object was sent on January 20, 2012, then a possible explicit date value would be January 23, 2012. The exact algorithm for determining a "best match" is implementation-specific.

3.1.4.4.1.2 Interpreting Relative Dates

Clients interpret the 15 least significant bits of the integer value generated in step 2 of the procedure specified in section 3.1.4.4.1 as a relative date by splitting the value into four values. The 2 most significant bits represent the modifier, the next 3 most significant bits represent the unit, the next 6 most significant bits represent the offset, and the 4 least significant bits represent the tag. These values are interpreted according to the following table.

Value	Meaning
Modifier	Specifies whether a relative date is early or late on the given day. If set to 00, there is no modifier. If set to 01, the modifier is "early". If set to 10, the modifier is "late".
Unit	Specifies the time unit used by the date. The possible values are 000 (Day), 001 (Week), 010 (Month), 011 (Year), 100 (Week of Month), and 101 (Day of Week).
Offset	Specifies the offset from the current date with respect to the unit specified in the unit value. This value MUST be interpreted as a signed value. Possible values are from -32 to 31.
Tag	The meaning of this value depends on the unit value. If the unit value is 100, the value of the tag indicates the month. If the unit is 101, the value of the tag indicates the day.

When the value of unit is 100, indicating "Week of Month", the value of the tag specifies the month. When the value of unit is 101, indicating "Day of Week", the value of the tag specifies the day, as specified in the following table.

Value of Tag	Meaning
0000	Sunday
0001	Monday

Value of Tag	Meaning
0010	Tuesday
0011	Wednesday
0100	Thursday
0101	Friday
0110	Saturday

The client SHOULD generate a date that reflects the next available date and time that matches the relative date based on the sent date and time of the **Message object**. For example, if the interpreted value is "next Wednesday" and the Message object was sent on August 16, 2012 (a Thursday), then a possible explicit date value would be August 22, 2012. The exact algorithm for determining a best match is implementation-specific.

3.1.4.5 Mail App Requests Web Services Identifier

When a **mail add-in** requests the web services identifier for an object, the client MUST derive the identifier using the following procedure:

1. If deriving a web services identifier for an **Email object**, a **Calendar object** that is not recurring, or a **Recurring Calendar object**, the client creates an **ItemData** structure as specified in section 2.2.6.1.1.1. If deriving a web services identifier for a single occurrence of a Recurring Calendar object, the client creates a **RecurrenceItemData** structure as specified in section 2.2.6.1.1.2. If deriving a web services identifier for a Conversation object, the client creates a **ConversationData** structure as specified in section 2.2.6.1.1.3.
2. The client creates a **DerivedId** structure as specified in section 2.2.6.1.1 and puts the structure created in step 1 in the **Data** field.
3. The client compresses the **DerivedId** structure using the following algorithm. For examples, see section 4.4.1 and section 4.4.2.
    1. If a byte value is repeated, replace the repeated bytes with three bytes. The repeated byte is written into the first two bytes, and the third byte is set to the total number of times that byte is repeated in the uncompressed stream, minus two.
    2. If a byte value is not repeated, copy it directly into the compressed stream.
4. If the length of the compressed **DerivedId** structure is greater than or equal to the length of the uncompressed **DerivedId** structure, clients MUST put the uncompressed **DerivedId** structure in the **Payload** field of a new **DerivedWSId** structure, as specified in section 2.2.6.1, and set the **CompressionType** field to 0x00. Otherwise, clients MUST put the compressed **DerivedId** structure in the **Payload** field of a new **DerivedWSId** structure and set the **CompressionType** field to 0x01.
5. The client encodes the **DerivedWSId** structure using **base64 encoding** and returns the result to the mail add-in.

3.1.5 Message Processing Events and Sequencing Rules

If the client receives a **RopNotify** **ROP response** ([MS-OXCROPS] section 2.2.14.2) for the event subscription specified in section 3.1.3, it SHOULD obtain a new copy of the **mail add-in** manifest using the **GetAppManifests** operation specified in [MS-OXWSCEXT] section 3.1.4.2.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

The server uses this protocol to notify clients of updates to the mail add-ins manifest and to add known entities to Message objects.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

The server uses this protocol to respond to the events specified in the following table.

Event	Details
The mail add-ins manifest is updated.	Section 3.2.4.1
New **Message objects** arrive in a **mailbox**.	Section 3.2.4.2

3.2.4.1 Mail App Manifest Updated

When the **mail add-in** manifest is updated, the server SHOULD send a **RopNotify** **ROP response** ([MS-OXCROPS] section 2.2.14.2) to all clients that have subscribed to a **TableModified** event on a table view that meets the following criteria:

- The table view is of the **FAI contents table** in the user's inbox.
- The table view includes items with a value of "IPM.Configuration.ExtensionMasterTable" in the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3).

3.2.4.2 New Message Object in Mailbox

When a new **Message object** arrives in a **mailbox**, the server SHOULD scan the contents of the Message object for known entities. The server SHOULD only scan **Email objects** and **Calendar objects**.

The server SHOULD scan Message object contents for the known entity types specified in [MS-OWEMXML]. The algorithm for scanning Message objects is implementation-specific and does not affect the operation of this protocol.

The server SHOULD create **XML documents** for each found known entity type as specified in section 2.2.3 (including subsections). The resulting XML documents SHOULD be stored in the known entity properties, using the mapping specified in the following table.

Known entity type	Known entity property
Address	**PidNameExtractedAddresses** (section 2.2.2.1)
Contact	**PidNameExtractedContacts** (section 2.2.2.2)
EmailAddress	**PidNameExtractedEmails** (section 2.2.2.3)
MeetingSuggestion	**PidNameExtractedMeetings** (section 2.2.2.4)
PhoneNumber	**PidNameExtractedPhones** (section 2.2.2.5)
TaskSuggestion	**PidNameExtractedTasks** (section 2.2.2.6)
Url	**PidNameExtractedUrls** (section 2.2.2.7)

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

The examples in section 4.1 through section 4.4 describe a scenario in which the client is hosting a single **mail add-in**.

4.1 Known Entities

4.1.1 Address Known Entity

The **mail add-in** requests the Address known entities on a message. The client checks the value of the **PidNameExtractedAddresses** property (section 2.2.2.1) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<AddressSet>
    <Version>15.0.0.0</Version>
    <Addresses>
        <Address StartIndex="1" Position="Subject">
            1234 Main St Buffalo, NY 98052
        </Address>
        <Address StartIndex="133" Position="Other">
            4567 1st St Seattle, WA 32008
        </Address>
    </Addresses>
</AddressSet>
```

4.1.2 Contact Known Entity

The **mail add-in** requests the Contact known entities on a message. The client checks the value of the **PidNameExtractedContacts** property (section 2.2.2.2) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<ContactSet>
    <Version>15.0.0.0</Version>
    <Contacts>
        <Contact>
            <Person StartIndex="63" Position="Other">
                <PersonString>Kim Akers</PersonString>
            </Person>
            <Phones>
                <Phone StartIndex="91" Position="Other">
                    <PhoneString>4255550102</PhoneString>
                    <OriginalPhoneString>425.555.0102</OriginalPhoneString>
                </Phone>
            </Phones>
            <Emails>
                <Email StartIndex="74" Position="Other">
                    <EmailString>kim@contoso.com</EmailString>
                </Email>
            </Emails>
            <ContactString>Kim Akers
kim@contoso.com
425.555.0102
            </ContactString>
        </Contact>
    </Contacts>
</ContactSet>
```

4.1.3 EmailAddress Known Entity

The **mail add-in** requests the EmailAddress known entities on a message. The client checks the value of the **PidNameExtractedEmails** property (section 2.2.2.3) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<EmailSet>
    <Version>15.0.0.0</Version>
    <Emails>
        <Email StartIndex="1032" Position="Other">
            <EmailString>jason@contoso.com</EmailString>
        </Email>
        <Email StartIndex="1058" Position="Signature">
            <EmailString>sanjay@contoso.com</EmailString>
        </Email>
    </Emails>
</EmailSet>
```

4.1.4 MeetingSuggestion Known Entity

The **mail add-in** requests the MeetingSuggestion known entities on a message. The client checks the value of the **PidNameExtractedMeetings** property (section 2.2.2.4) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<MeetingSet>
    <Version>15.0.0.0</Version>
    <Meetings>
        <Meeting Location="My office" Subject="Project Status"
            StartIndex="56" Position="LatestReply">
            <MeetingString>
                Let's meet tomorrow at 3pm in my office to discuss the project.
            </MeetingString>
            <Attendees>
                <EmailUser Id="sanjay@contoso.com">Sanjay Shah</EmailUser>
            </Attendees>
            <StartTime>2012-03-10T23:00:00Z</StartTime>
            <EndTime>2012-03-10T23:30:00Z</EndTime>
        </Meeting>
    </Meetings>
</MeetingSet>
```

4.1.4.1 Interpreting a Precise Date Value

The **mail add-in** requests the MeetingSuggestion known entities on a message. The client checks the value of the **PidNameExtractedMeetings** property (section 2.2.2.4) and finds the following value for the **StartTime** element, as described in section 2.2.3.1.29.

```
0040-01-13T20:00:00Z
```

Because January 13, 0040, is before June 15, 1436, the value is an encoded date. The client interprets the value according to the rules described in section 3.1.4.4.1.

1. The client determines that the number of days from January 1, 0001, to January 13, 0040, is 14256.
2. In 18-bit binary form, this value is 000011011110110000.
3. The 3 most significant bits are 000, which indicates that the encoded date is a precise date.

Because this is a precise date, the 15 least significant bits (011011110110000) are interpreted according to the rules described in section 3.1.4.4.1.1.

1. The value is split into two values. The 3 most significant bits (011) represent the type, and the 12 remaining bits (011110110000) represent the value**.**

2. Since the type bits are 011, the value represents a month and day. The 4 most significant bits (0111) are converted to decimal (7) to determine the month (July). The next 5 most significant bits (10110) are converted to decimal (22) to determine the day.

The precise date value is July 22.

4.1.4.2 Interpreting a Relative Date Value

The **mail add-in** requests the MeetingSuggestion known entities on a message. The client checks the value of the **PidNameExtractedMeetings** property (section 2.2.2.4) and finds the following value for the **StartTime** element, as described in section 2.2.3.1.29.

```
0127-03-19T22:30:00Z
```

Because March 19, 0127, is before June 15, 1436, the value is an encoded date. The client interprets the value according to the rules described in section 3.1.4.4.1.

1. The client determines that the number of days from January 1, 0001, to March 19, 0127, is 46097.

2. In 18-bit binary form, this value is 001011010000010001.

3. The 3 most significant bits are 001, which indicates that the encoded date is a relative date.

Because this is a relative date, the 15 least significant bits (011010000010001) are interpreted according to the rules described in section 3.1.4.4.1.2.

1. The value is split into four values. The 2 most significant bits (01) represent the modifier, the next 3 most significant bits (101) represent the unit, the next 6 most significant bits (000001) represent the offset, and the remaining 4 bits (0001) represent the tag.

2. The value of the modifier indicates that a modifier of "early" is applied.

3. The value of the unit indicates that the unit is Day of Week.

4. The value of the offset is 1, meaning the next instance of the indicated day of the week.

5. The value of the tag indicates that the day of the week is Monday.

The relative date value is "early next Monday".

4.1.5 PhoneNumber Known Entity

The **mail add-in** requests the PhoneNumber known entities on a message. The client checks the value of the **PidNameExtractedPhones** property (section 2.2.2.5) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<PhoneSet>
    <Version>15.0.0.0</Version>
    <Phones>
        <Phone StartIndex="16" Position="LatestReply">
            <PhoneString>4255550100</PhoneString>
            <OriginalPhoneString>(425)555-0100</OriginalPhoneString>
```

```
        </Phone>
        <Phone StartIndex="942" Position="Other">
            <PhoneString>4255550101</PhoneString>
            <OriginalPhoneString>(425) 555 0101</OriginalPhoneString>
        </Phone>
    </Phones>
</PhoneSet>
```

4.1.6 TaskSuggestion Known Entity

The **mail add-in** requests the TaskSuggestion known entities on a message. The client checks the value of the **PidNameExtractedTasks** property (section 2.2.2.6) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<TaskSet>
    <Version>15.0.0.0</Version>
    <Tasks>
        <Task StartIndex="42" Position="LatestReply">
            <TaskString>Please send a copy of the presentation to Bob.</TaskString>
            <Assignees>
                <EmailUser Id="jason@contoso.com">Jason Carlson</EmailUser>
            </Assignees>
        </Task>
    </Tasks>
</TaskSet>
```

4.1.7 Url Known Entity

The **mail add-in** requests the Url known entities on a message. The client checks the value of the **PidNameExtractedUrls** property (section 2.2.2.7) and finds the following **XML document**.

```
<?xml version="1.0" encoding="utf-16"?>
<UrlSet>
    <Version>15.0.0.0</Version>
    <Urls>
        <Url StartIndex="252" Position="LatestUrl" Type="Url">
            <UrlString>http://www.contoso.com/</UrlString>
        </Url>
        <Url StartIndex="378" Position="Signature" Type="Url">
            <UrlString>https://www.contoso.com/img/companylog.jpg</UrlString>
        </Url>
    </Urls>
</UrlSet>
```

4.2 Mail App Configuration Data

The **mail add-in** requests access to its configuration data. The identifier for the mail add-in is "urn:uuid:{4b8686f0-1b40-11e1-bddb-0800200c9a66}".

The client opens the configuration data message with the value "IPM.Configuration.ClientExtension.4b8686f01b4011e1bddb0800200c9a66" in the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3). The client checks the value of the **PidTagRoamingDictionary** property ([MS-OXOCFG] section 2.2.2.2) on the configuration data message and finds the following **XML document**.

```
<?xml version="1.0"?>
<UserConfiguration>
```

```
    <Info version="Outlook.15"/>
    <Data>
        <e k="18-ExtensionSettings" v="18-
{&quot;application_setting_name_1&quot;:&quot;\&quot;application_setting_1\&quot;&quot;,&quot
;application_setting_name_2&quot;:&quot;\&quot;application_setting_2\&quot;&quot;,&quot;appli
cation setting name 3&quot;:&quot;\&quot;application setting 3\&quot;&quot;}"/>
        <e k="18-OLPrefsVersion" v="9-1"/>
    </Data>
</UserConfiguration>
```

The value of the "ExtensionSettings" name/value pair is extracted as follows.

```
{"application_setting_name_1":"\"application_setting_1\"",
 "application setting name 2":"\"application setting 2\"",
 "application_setting_name_3":"\"application_setting_3\""}
```

4.3 Mail App Custom Properties

The **mail add-in** requests access to its custom properties on a message. The identifier for the mail add-in is "urn:uuid:{4C13B122-C256-47B0-A4BF-9ABBFE396473}".

The client uses the **property name** "cecp-4c13b122-c256-47b0-a4bf-9abbfe396473" and the PS_PUBLIC_STRINGS **property set** to generate a **property tag**. Using this property tag, the client checks the value of the property and finds the following **JSON** object.

```
{"custom_property_name_1":"custom_property_1",
 "custom_property_name_2":"custom_property_2",
 "custom_property_name_3":"custom_property_3"}
```

4.4 Derived Web Services Identifier

The following examples illustrate deriving a web services identifier for an **Email object** (section 4.4.1) and for a single occurrence of a **Recurring Calendar object** (section 4.4.2).

4.4.1 Derived Web Services Identifier for an Email Object

The **mail add-in** requests the web services identifier for an **Email object** with the following value for its **Message EntryID** structure, as described in [MS-OXCDATA] section 2.2.4.2.

```
0000000088E6E5A0C938724DB22D21E35B7BEF6107008CE5522DEFA36348B3A449578E1E6774000000022742000008
CE5522DEFA36348B3A449578E1E67740000000235400000
```

The **GUID** for the **mailbox** is 6123e271-3ea9-4de3-a56e-90172eff4539.

The client sets the fields in a **DerivedId** structure, as described in section 2.2.6.1.1, as shown in the following table.

DerivedId field	Value
Reserved	0x03
MailboxGuidSize	0x0024
MailboxGuid	36313233653237312D336561392D34646533 2D613536652D393031373265666634353339

DerivedId field	Value
ObjectType	0x00
Data	**ItemData** structure (section 2.2.6.1.1.1)

The fields of the **ItemData** structure in the **Data** field are set as shown in the following table.

ItemData field	Value
EntryIdSize	0x0046
EntryId	0000000088E6E5A0C938724DB22D21E35B7B EF6107008CE5522DEFA36348B3A449578E1E6 774000000022742000008CE5522DEFA36348B3 A449578E1E67740000000235400000

The structure is represented by the following bytes.

```
0324003631323365323731 2D336561392D346465332D613536652D393031373265666634353339004600000000008
8E6E5A0C938724DB22D21E35B7BEF6107008CE5522DEFA36348B3A449578E1E6774000000022742000008CE5522DEF
A36348B3A449578E1E67740000000235400000
```

Compressing these bytes as described in section 3.1.4.5 results in the following bytes.

```
0324003631323365323731 2D336561392D346465332D613536652D393031373265666600343533390046000003 88E
6E5A0C938724DB22D21E35B7BEF6107008CE5522DEFA36348B3A449578E1E6774000001022742000008CE5522DEF
A36348B3A449578E1E6774000001023540000000
```

Because the length of the compressed structure is greater than the length of the uncompressed structure, the client puts the uncompressed structure in the **Payload** field of the **DerivedWSId** structure, as described in section 2.2.6.1, and sets the **CompressionType** field to 0x00. The client then encodes the data with **base64 encoding** to generate the following web services identifier.

```
AAMkADYxMjNlMjcxLTNlYTktNGRlMy1hNTZlLTkwMTcyZWZmNDUzOQBGAAAAAACI5uWgyThyTbItIeNbe+9hBwCM5VIt7
6NjSLOkSVeOHmd0AAAAAidCAACM5VIt76NjSLOkSVeOHmd0AAAAAjVAAAA=
```

4.4.2 Derived Web Services Identifier for a Single Occurrence

The **mail add-in** requests the web services identifier for a single occurrence of a **Recurring Calendar object** with the following value for its **Message EntryID** structure, as described in [MS-OXCDATA] section 2.2.4.2.

```
00000000608903A1BC65744E80B444444444EC0307000F43FB93C5EBC841B4AE3351F9FD201800000000000F00000
F43FB93C5EBC841B4AE3351F9FD20180000000007F30000
```

The **GUID** for the **mailbox** is 6123e271-3ea9-4de3-a56e-90172eff4539.

The start date of the occurrence is March 13, 2012.

The client sets the fields in a **DerivedId** structure, as described in section 2.2.6.1.1, as shown in the following table.

DerivedId field	Value
Reserved	0x03
MailboxGuidSize	0x0024
MailboxGuid	36313233653237312D336561392D34646533 2D613536652D393031373265666634353339
ObjectType	0x01
Data	**RecurrenceItemData** structure (section 2.2.6.1.1.2)

The fields of the **RecurrenceItemData** structure in the **Data** field are set as shown in the following table.

RecurrenceItemData field	Value
Size	0x0051
DateSize	0x08
Date	0x08CECEC35E308000
EntryIdSize	0x46
EntryId	00000000608903A1BC65744E80B444444444 EC0307000F43FB93C5EBC841B4AE3351F9FD 2018000000000000F00000F43FB93C5EBC841 B4AE3351F9FD20180000000007F30000
Reserved	0x10

The structure is represented by the following bytes.

```
03240036313233653237312D336561392D346465332D613536652D393031373265666634353339015100080CECEC
35E308000460000000608903A1BC65744E80B444444444EC0307000F43FB93C5EBC841B4AE3351F9FD2018000000
00000F00000F43FB93C5EBC841B4AE3351F9FD20180000000007F3000010
```

Compressing these bytes as described in section 3.1.4.5 results in the following bytes.

```
03240036313233653237312D336561392D346465332D613536652D393031373265666600343533390151000808000C
ECE00C35E308000460000002608903A1BC65744E80B4444402EC0307000F43FB93C5EBC841B4AE3351F9FD20180000
030F0000000F43FB93C5EBC841B4AE3351F9FD201800000207F300000010
```

Because the length of the compressed structure is equal to the length of the uncompressed structure, the client puts the uncompressed structure in the **Payload** field of the **DerivedWSId** structure, as described in section 2.2.6.1, and sets the **CompressionType** field to 0x00. The client then encodes the data with **base64 encoding** to generate the following web services identifier.

```
AAMkADYxMjNlMjcxLTNlYTktNGRlMy1hNTZlLTkwMTcyZWZmNDUzOQFRAAgIzs7DXjCAAEYAAAAAYIkDobxldE6AtERER
ETsAwcAD0P7k8XryEG0rjNR+f0gGAAAAAAADwAAD0P7k8XryEG0rjNR+f0gGAAAAAAH8wAAEA==
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full XML Schema

For ease of implementation, the following is the full **XML schema** for this protocol.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TaskSet" nillable="true" type="TaskSet" />
  <xs:complexType name="TaskSet">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Version"
        type="Version" />
      <xs:element minOccurs="0" maxOccurs="1" name="Tasks"
        type="ArrayOfTask" />
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="Version">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]+\.[0-9]+(\.[0-9]+(\.[0-9]+)?)?"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ArrayOfTask">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Task"
        nillable="true" type="Task" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Task">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="TaskString"
        type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1" name="Assignees"
        type="ArrayOfEmailUser" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position"
        type="EmailPosition" />
  </xs:complexType>
  <xs:complexType name="ArrayOfEmailUser">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="EmailUser"
        nillable="true" type="EmailUser" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EmailUser">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Id" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:simpleType name="EmailPosition">
    <xs:restriction base="xs:string">
      <xs:enumeration value="LatestReply" />
      <xs:enumeration value="Subject" />
      <xs:enumeration value="Signature" />
      <xs:enumeration value="Other" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="AddressSet" nillable="true" type="AddressSet" />
  <xs:complexType name="AddressSet">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Version"
        type="Version" />
      <xs:element minOccurs="0" maxOccurs="1" name="Addresses"
        type="ArrayOfAddress" />
    </xs:sequence>
```

```
</xs:complexType>
<xs:complexType name="ArrayOfAddress">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Address"
      nillable="true" type="Address" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Address">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute default="-1" name="StartIndex" type="xs:int" />
      <xs:attribute default="LatestReply" name="Position"
        type="EmailPosition" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="MeetingSet" nillable="true" type="MeetingSet" />
<xs:complexType name="MeetingSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Version"
      type="Version" />
    <xs:element minOccurs="0" maxOccurs="1" name="Meetings"
      type="ArrayOfMeeting" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfMeeting">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Meeting"
      nillable="true" type="Meeting" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Meeting">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="MeetingString"
      type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="Attendees"
      type="ArrayOfEmailUser" />
    <xs:element minOccurs="1" maxOccurs="1" name="StartTime"
      nillable="true" type="xs:dateTime" />
    <xs:element minOccurs="1" maxOccurs="1" name="EndTime" nillable="true"
      type="xs:dateTime" />
  </xs:sequence>
  <xs:attribute name="Location" type="xs:string" />
  <xs:attribute name="Subject" type="xs:string" />
  <xs:attribute default="-1" name="StartIndex" type="xs:int" />
  <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
<xs:element name="PhoneSet" nillable="true" type="PhoneSet" />
<xs:complexType name="PhoneSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Version"
      type="Version" />
    <xs:element minOccurs="0" maxOccurs="1" name="Phones"
      type="ArrayOfPhone" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfPhone">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Phone"
      nillable="true" type="Phone" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Phone">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="PhoneString"
      type="xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="OriginalPhoneString"
      type="xs:string" />
  </xs:sequence>
```

```
  <xs:attribute default="-1" name="StartIndex" type="xs:int" />
  <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
  <xs:attribute default="Unspecified" name="Type" type="PhoneType" />
</xs:complexType>
<xs:simpleType name="PhoneType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Unspecified" />
    <xs:enumeration value="Home" />
    <xs:enumeration value="Mobile" />
    <xs:enumeration value="Work" />
    <xs:enumeration value="Fax" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="EmailSet" nillable="true" type="EmailSet" />
<xs:complexType name="EmailSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Version"
      type="Version" />
    <xs:element minOccurs="0" maxOccurs="1" name="Emails"
      type="ArrayOfEmail" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfEmail">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Email"
      nillable="true" type="Email" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Email">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="EmailString"
      type="xs:string" />
  </xs:sequence>
  <xs:attribute default="-1" name="StartIndex" type="xs:int" />
  <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
</xs:complexType>
<xs:element name="UrlSet" nillable="true" type="UrlSet" />
<xs:complexType name="UrlSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Version"
      type="Version" />
    <xs:element minOccurs="0" maxOccurs="1" name="Urls"
      type="ArrayOfUrl" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfUrl">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Url"
      nillable="true" type="Url" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Url">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="UrlString"
      type="xs:string" />
  </xs:sequence>
  <xs:attribute default="-1" name="StartIndex" type="xs:int" />
  <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
  <xs:attribute default="Unspecified" name="Type" type="UrlType" />
</xs:complexType>
<xs:simpleType name="UrlType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Unspecified" />
    <xs:enumeration value="Url" />
    <xs:enumeration value="Filename" />
  </xs:restriction>
</xs:simpleType>
<xs:element name="ContactSet" nillable="true" type="ContactSet" />
<xs:complexType name="ContactSet">
```

```
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Version"
        type="Version" />
      <xs:element minOccurs="0" maxOccurs="1" name="Contacts"
        type="ArrayOfContact" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfContact">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Contact"
        nillable="true" type="Contact" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Person">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="PersonString"
        type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
  </xs:complexType>
  <xs:complexType name="Business">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="BusinessString"
        type="xs:string" />
    </xs:sequence>
    <xs:attribute default="-1" name="StartIndex" type="xs:int" />
    <xs:attribute default="LatestReply" name="Position" type="EmailPosition" />
  </xs:complexType>
  <xs:complexType name="Contact">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1" name="Person" type="Person" />
      <xs:element minOccurs="0" maxOccurs="1" name="Business"
        type="Business" />
      <xs:element minOccurs="0" maxOccurs="1" name="Phones"
        type="ArrayOfPhone" />
      <xs:element minOccurs="0" maxOccurs="1" name="Urls"
        type="ArrayOfUrl" />
      <xs:element minOccurs="0" maxOccurs="1" name="Emails"
        type="ArrayOfEmail" />
      <xs:element minOccurs="0" maxOccurs="1" name="Addresses"
        type="ArrayOfAddress" />
      <xs:element minOccurs="0" maxOccurs="1" name="ContactString"
        type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

7 Appendix B: Product Behavior

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

- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Exchange Server 2019
- Microsoft Outlook 2019
- Microsoft Outlook 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index