

[MS-AEMAIL]: ActiveSync E-Mail Class Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0.0	Major	Initial Release.
02/04/2009	1.0.1	Editorial	Revised and edited technical content.
03/04/2009	1.0.2	Editorial	Revised and edited technical content.
04/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
07/15/2009	3.0.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	5.0.0	Major	Updated and revised the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary.....	6
1.2	References.....	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Protocol Overview	7
1.4	Relationship to Other Protocols.....	7
1.5	Prerequisites/Preconditions.....	7
1.6	Applicability Statement.....	8
1.7	Versioning and Localization	8
1.8	Vendor-Extensible Fields	8
1.9	Standards Assignments	8
2	Messages	9
2.1	Transport.....	9
2.2	Message Syntax.....	9
2.2.1	Complex Types.....	12
2.2.1.1	Attachments.....	12
2.2.1.2	Attachments.Attachment.....	12
2.2.1.3	Body.....	13
2.2.1.4	MeetingRequest	13
2.2.1.5	MeetingRequest.Recurrences	13
2.2.1.6	MeetingRequest.Recurrences.Recurrence	13
2.2.1.7	Flag	13
2.2.1.8	Categories.....	13
2.2.2	Elements.....	14
2.2.2.1	To	17
2.2.2.2	Cc.....	17
2.2.2.3	From.....	17
2.2.2.4	Subject.....	17
2.2.2.5	ReplyTo.....	17
2.2.2.6	DateReceived.....	17
2.2.2.7	DisplayTo.....	18
2.2.2.8	ThreadTopic	18
2.2.2.9	Importance	18
2.2.2.10	Read	18
2.2.2.11	Attachments.Attachment.DisplayName	18
2.2.2.12	Attachments.Attachment.UmAttOrder.....	18
2.2.2.13	Attachments.Attachment.UmAttDuration	19
2.2.2.14	MessageClass.....	19
2.2.2.15	MeetingRequest.AllDayEvent	20
2.2.2.16	MeetingRequest.StartTime.....	20
2.2.2.17	MeetingRequest.DtStamp	21
2.2.2.18	MeetingRequest.EndTime	21
2.2.2.19	MeetingRequest.InstanceType	21
2.2.2.20	MeetingRequest.Location.....	21
2.2.2.21	MeetingRequest.Organizer.....	21
2.2.2.22	MeetingRequest.RecurrenceId	21
2.2.2.23	MeetingRequest.Reminder	22
2.2.2.24	MeetingRequest.ResponseRequested	22

2.2.2.25	MeetingRequest.Recurrences.Recurrence.Type.....	22
2.2.2.26	MeetingRequest.Recurrences.Recurrence.Interval.....	22
2.2.2.27	MeetingRequest.Recurrences.Recurrence.Until	22
2.2.2.28	MeetingRequest.Recurrences.Recurrence.Occurrences	23
2.2.2.29	MeetingRequest.Recurrences.Recurrence.WeekOfMonth	23
2.2.2.30	MeetingRequest.Recurrences.Recurrence.DayOfMonth	23
2.2.2.31	MeetingRequest.Recurrences.Recurrence.DayOfWeek	23
2.2.2.32	MeetingRequest.Recurrences.Recurrence.MonthOfYear	24
2.2.2.33	MeetingRequest.Recurrences.Recurrence.CalendarType	24
2.2.2.34	MeetingRequest.Recurrences.Recurrence.IsLeapMonth.....	25
2.2.2.35	MeetingRequest.Sensitivity.....	25
2.2.2.36	MeetingRequest.BusyStatus.....	25
2.2.2.37	MeetingRequest.TimeZone.....	26
2.2.2.38	MeetingRequest.GlobalObjId.....	26
2.2.2.39	MeetingRequest.DisallowNewTimeProposal	26
2.2.2.40	InternetCPID.....	26
2.2.2.41	Flag.Subject.....	26
2.2.2.42	Flag.Status.....	26
2.2.2.43	Flag.FlagType.....	27
2.2.2.44	Flag.DateCompleted	27
2.2.2.45	Flag.CompleteTime.....	27
2.2.2.46	Flag.StartDate.....	27
2.2.2.47	Flag.DueDate	28
2.2.2.48	Flag.UTCStartDate.....	28
2.2.2.49	Flag.UTCDueDate	28
2.2.2.50	Flag.ReminderSet.....	28
2.2.2.51	Flag.ReminderTime	29
2.2.2.52	Flag.OrdinalDate	29
2.2.2.53	Flag.SubOrdinalDate.....	29
2.2.2.54	NativeBodyType	29
2.2.2.55	ContentClass.....	29
2.2.2.56	UmCallerID	29
2.2.2.57	UmUserNotes	30
2.2.2.58	ConversationId.....	30
2.2.2.59	ConversationIndex	31
2.2.2.60	LastVerbExecuted.....	31
2.2.2.61	LastVerbExecutionTime	32
2.2.2.62	ReceivedAsBcc	32
2.2.2.63	Sender.....	32
2.2.2.64	Categories.Category.....	32

3	Protocol Details.....	33
3.1	Client Details.....	33
3.1.1	Abstract Data Model.....	33
3.1.2	Timers	33
3.1.3	Initialization	33
3.1.4	Higher-Layer Triggered Events	33
3.1.4.1	Synchronize E-mail Between Client and Server.....	33
3.1.4.2	Search E-mail	33
3.1.4.3	Retrieve Individual E-mail	33
3.1.4.4	Send Flag Updates to the Server	33
3.1.5	Message Processing Events and Sequencing Rules	35
3.1.5.1	ItemOperations Command Request	35

3.1.5.2	Search Command Request.....	36
3.1.5.3	Sync Command Request.....	36
3.1.6	Timer Events.....	36
3.1.7	Other Local Events	36
3.2	Server Details	36
3.2.1	Abstract Data Model.....	36
3.2.2	Timers	37
3.2.3	Initialization	37
3.2.4	Higher-Layer Triggered Events	37
3.2.4.1	Synchronize E-mail Between Client and Server.....	37
3.2.4.2	Search E-mail	37
3.2.4.3	Retrieve Individual E-mail	37
3.2.4.4	Change Tracking Semantics for Flagging Properties	37
3.2.4.5	Send Changes to the Client	37
3.2.5	Message Processing Events and Sequencing Rules	38
3.2.5.1	ItemOperations Command Response	39
3.2.5.2	Search Command Response	39
3.2.5.3	Sync Command Response	39
3.2.6	Timer Events.....	39
3.2.7	Other Local Events	39
4	Protocol Examples	40
4.1	Synchronizing E-Mail.....	40
4.1.1	Example Sync Request for Inbox with Metadata	40
4.1.2	Example Sync Response for Inbox with Metadata.....	40
4.1.3	Example Sync Request for Inbox with Body Preferences.....	41
4.1.4	Example Sync Response Adding One HTML Message	42
4.1.5	Example Sync Response Adding an Electronic Voice Mail Attachment	43
4.1.6	Example Sync Response Adding a Text Attachment	44
4.1.7	Example Sync Request Deleting One E-mail.....	45
4.1.8	Example Success Sync Response.....	46
4.1.9	Example Sync Response Adding a Meeting Request	46
4.1.10	Example Sync Response Adding a Meeting Request with a Recurrence	47
4.1.11	Example ItemOperations Request.....	49
4.1.12	Example ItemOperations Response With Fetched Content.....	49
4.2	Setting MeetingRequest Recurrence Intervals	50
4.3	Setting Flags on the Client and Server.....	51
4.3.1	Example Sync Request Setting a Flag on the Client.....	51
4.3.2	Example Sync Request Setting a Flag on the Server	52
4.3.3	Example Sync Request Setting the Complete Flag.....	53
4.3.4	Example Sync Request Clearing a Flag on the Client	53
5	Security.....	55
5.1	Security Considerations for Implementers.....	55
5.2	Index of Security Parameters	55
6	Appendix A: Product Behavior	56
7	Change Tracking	58
8	Index.....	64

1 Introduction

This document specifies the XML representation of e-mail data sent or received on mobile devices that communicate by using the ActiveSync protocols.

1.1 Glossary

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

binary large object (BLOB)
class
code page
collection
conversation
Coordinated Universal Time (UTC)
delivery receipt
Hypertext Markup Language (HTML)
message database (MDB)
non-delivery report (NDR)
non-read receipt
read receipt
recipient (2)
synchronization
Uniform Resource Identifier (URI)
WAP Binary XML (WBXML)
XML

The following terms are specific to this document:

Voice over Internet Protocol (VoIP): The transmission of voice capabilities over the Internet.

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

1.2 References

1.2.1 Normative References

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

[E164] International Telecom Union, "The international public telecommunication numbering plan", February 2005, <http://www.itu.int/rec/T-REC-E.164-200502-I/en>.

[MS-ASAIRS] Microsoft Corporation, "[ActiveSync AirSyncBase Namespace Protocol Specification](#)", December 2008.

[MS-ASCMD] Microsoft Corporation, "[ActiveSync Command Reference Protocol Specification](#)", December 2008.

[MS-ASDTYPE] Microsoft Corporation, "[ActiveSync Data Types](#)", December 2008.

[MS-ASWBXML] Microsoft Corporation, "[ActiveSync WAP Binary XML \(WBXML\) Protocol Specification](#)", December 2008.

[MS-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, <http://go.microsoft.com/fwlink/?LinkId=111558>.

[MS-IPFFX] Microsoft Corporation, "InfoPath Form File Format Specification", June 2008, <http://go.microsoft.com/fwlink/?LinkId=148970>.

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

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

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002, <http://www.ietf.org/rfc/rfc3261.txt>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008, <http://www.ietf.org/rfc/rfc5322.txt>.

[XML] Bray, T., et al., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", November 2008, <http://www.w3.org/TR/REC-xml/>.

1.2.2 Informative References

None.

1.3 Protocol Overview

This document specifies the XML representation of e-mail data sent or received on mobile devices that communicate by using the ActiveSync protocols. E-mail data is included in protocol command requests when e-mail data is being sent from the client to the server. E-mail data is also included in protocol command responses when e-mail data is retrieved from the server. E-mail data includes header information such as to, from, and subject, as well as body, attachment, flag, and meeting request information.

1.4 Relationship to Other Protocols

This document specifies the XML representation of e-mail message data that is sent and received by the protocol commands, as specified in [\[MS-ASCMD\]](#).

All data types in this document conform to the data type definitions specified in [\[MS-ASDTYPE\]](#).

Estimated data size, body content, and data truncation information about e-mail messages are not part of the E-mail **class** data. Instead, that data is contained in the AirSyncBase namespace, as specified in [\[MS-ASAIRS\]](#).

The **code page** used to encode E-mail class data is specified in [\[MS-ASWBXML\]](#).

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol specifies a set of elements and complex types for use in communicating e-mail data using the commands specified in [\[MS-ASCMD\]](#). This set of elements and complex types is applicable when communicating e-mail data such as to, from, and subject, as well as body, attachment, flag, and meeting request information between a mobile device and a server. These elements and complex types are not applicable when sending calendar, task, note, or contact data between a mobile device and a server.

1.7 Versioning and Localization

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The E-mail class consists of a series of XML types and elements that are embedded inside of a command request or response. The complex types and elements of the E-mail class are defined in three namespaces: Email, Email2<1>, and AirSyncBase. All of the E-mail class complex types and elements are specified in this document. However, complex types and elements defined in the AirSyncBase namespace are further specified in [\[MS-ASAIRS\]](#).

2.2 Message Syntax

The markup MUST be well-formed XML, as specified in [\[XML\]](#) section 2.1, using the commands specified in [\[MS-ASCMD\]](#).

The XML markup that constitutes the request body or the response body is transmitted between the client and server using **WAP Binary XML (WBXML)** [\[MS-ASWBXML\]](#).

The XML schema definition for the E-mail class is as follows. This schema represents the full set of data returned by the **Ssync** command. Note that some of the elements are imported from the Tasks, AirSyncBase and Email2 namespaces.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="EMAIL:" attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="EMAIL:" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:A="AirSyncBase:"
  xmlns:email2="Email2" xmlns:tasks="Tasks:">
  <xs:import namespace="AirSyncBase" />
  <xs:import namespace="Tasks:" />
  <xs:import namespace="Email2:" />
  <xs:element name="To" minOccurs="0" maxOccurs="1" >
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32768" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Cc" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="From" minOccurs="0" maxOccurs="1" >
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32768" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Subject" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="ReplyTo" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="DateReceived" type="xs:dateTime" minOccurs="0" maxOccurs="1" />
  <xs:element name="DisplayTo" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="ThreadTopic" type="xs:string" minOccurs="0" maxOccurs="1" />
  <xs:element name="Importance" type="xs:unsignedByte" minOccurs="0" maxOccurs="1" />
  <xs:element name="Read" type="xs:boolean" minOccurs="0" maxOccurs="1" />
  <xs:element name="Attachments" type="A:Attachments" minOccurs="0" maxOccurs="1" >
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Attachment" minOccurs="1" maxOccurs="unbounded" >
          <xs:complexType>
```

```

        <xs:sequence>
            <xs:element name="DisplayName" type="xs:string" minOccurs="0"
maxOccurs="1" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Body" type="A:Body" minOccurs="0" maxOccurs="1"/>
<xs:element name="MessageClass" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="MeetingRequest" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AllDayEvent" type="xs:unsignedByte" minOccurs="1"
maxOccurs="1" />
            <xs:element name="StartTime" type="xs:dateTime" minOccurs="1"
maxOccurs="1" />
            <xs:element name="DtStamp" type="xs:dateTime" minOccurs="1" maxOccurs="1"
/>
            <xs:element name="EndTime" type="xs:dateTime" minOccurs="1" maxOccurs="1"
/>
            <xs:element name="InstanceType" type="xs:unsignedByte" minOccurs="1"
maxOccurs="1" />
            <xs:element minOccurs="0" name="Location" minOccurs="0" maxOccurs="1" >
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="32768" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Organizer" type="xs:string" minOccurs="0" maxOccurs="1"
/>
            <xs:element minOccurs="0" name="RecurrenceId" type="xs:dateTime"
minOccurs="0" maxOccurs="1" />
            <xs:element name="Reminder" type="xs:unsignedShort" minOccurs="0"
maxOccurs="1" />
            <xs:element name="ResponseRequested" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1" />
            <xs:element name="Recurrences" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Recurrence" minOccurs="1" maxOccurs="1">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Type"
type="xs:unsignedByte" minOccurs="1" maxOccurs="1" />
                                    <xs:element name="Interval"
type="xs:integer" minOccurs="1" maxOccurs="1" />
                                    <xs:element name="Until" type="xs:datetime"
minOccurs="1" maxOccurs="1" />
                                    <xs:element name="Occurrences"
type="xs:integer" minOccurs="1" maxOccurs="1" />
                                    <xs:element name="WeekOfMonth"
type="xs:integer" minOccurs="0" maxOccurs="1" />
                                    <xs:element name="DayOfMonth"
type="xs:integer" minOccurs="0" maxOccurs="1" />
                                    <xs:element name="DayOfWeek"
type="xs:integer" minOccurs="0" maxOccurs="1" />
                                    <xs:element name="MonthOfYear"
type="xs:integer" minOccurs="0" maxOccurs="1" />
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

type="xs:integer" minOccurs="0" maxOccurs="1" />
<xs:element name="CalendarType"
type="xs:unsignedByte" minOccurs="0" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Sensitivity" type="xs:integer" minOccurs="0"
maxOccurs="1" />
<xs:element name="BusyStatus" type="xs:integer" minOccurs="0"
maxOccurs="1" />
<xs:element name="TimeZone" type="xs:string" minOccurs="1" maxOccurs="1"
/>
<xs:element name="GlobalObjId" type="xs:string" minOccurs="1"
maxOccurs="1" />
<xs:element name="DisallowNewTimeProposal" type="xs:unsignedByte"
minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="InternetCPID" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="Flag" minOccurs="0" maxOccurs="1" >
<xs:complexType>
<xs:sequence>
<xs:element ref="tasks:Subject" type="xs:string" minOccurs="0"
maxOccurs="1" />
<xs:element name="Status" type="xs:integer" minOccurs="0" maxOccurs="1"
/>
<xs:element name="FlagType" type="xs:string" minOccurs="0" maxOccurs="1"
/>
<xs:element ref="tasks:DateCompleted" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element name="CompleteTime" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:StartDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:DueDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:UTCStartDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:UTCDueDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:ReminderSet" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:ReminderTime" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:OrdinalDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="tasks:SubOrdinalDate" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ContentClass" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="NativeBodyType" type="A:NativeBodyType" minOccurs="0" maxOccurs="1" />
<xs:element ref="email2:UmCallerID" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element ref="email2:UmUserNotes" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element ref="email2:UmAttOrder" type="xs:integer" minOccurs="0" maxOccurs="1" />

```

```

<xs:element ref="email2:UmAttDuration" type="xs:integer" minOccurs="0" maxOccurs="1" />
<xs:element ref="email2:ConversationId" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element ref="email2:ConversationIndex" type="xs:string" minOccurs="1" maxOccurs="1"
/>
<xs:element ref="email2:LastVerbExecuted" type="xs:integer" minOccurs="0" maxOccurs="1"
/>
<xs:element ref="email2:LastVerbExecutionTime" type="xs:dateTime" minOccurs="0"
maxOccurs="1" />
<xs:element ref="email2:ReceivedAsBcc" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1" />
<xs:element ref="email2:Sender" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element ref="email2:Categories" type="xs:string" minOccurs="0" maxOccurs="1" >
  <xs:complexType>
    <xs:element ref="email2:Category" type="xs:string" minOccurs="0"
maxOccurs="300" />
  </xs:complexType>
</xs:element>
</xs:schema>

```

2.2.1 Complex Types

The following table summarizes the set of common XML schema complex types defined by this specification.

Complex Type	Description
Attachments	The collection of Attachment elements.
Attachments.Attachment	The e-mail attachment.
Body	A description of the body text, along with its data.
MeetingRequest	A meeting request accompanying an e-mail message.
MeetingRequest.Recurrences	A collection of Recurrence elements.
MeetingRequest.Recurrences.Recurrence	A collection of Recurrence elements that describe when and how often this meeting recurs.
Flag	The flag associated with the item, along with the item's current status.
Categories	The user-selected categories for this message.

2.2.1.1 Attachments

The **Attachments** type is an optional **container** ([\[MS-ASDTYPE\]](#) section 2.8) type that contains a collection of attachment elements.

If an **Attachment** type is defined, then it MUST contain one or more instances of the **Attachments.Attachment** type. The attachments type is part of the AirSyncBase namespace, and is further specified in [\[MS-ASAIRS\]](#) section 2.2.1.1.

2.2.1.2 Attachments.Attachment

The **Attachment** type is a **container** type that represents an e-mail attachment.

If an attachments type is defined, it MUST contain one or more instances of the **Attachments.Attachment** type.

The **Attachment** type is part of the AirSyncBase namespace, and is further specified in [\[MS-ASAIRS\]](#) section 2.2.1.2.

2.2.1.3 Body

The **Body** type is an optional **container** type that contains the message text of the e-mail, along with associated message body data.

The **Body** type is part of the AirSyncBase namespace, and is further specified in [\[MS-ASAIRS\]](#) section 2.2.1.3.

2.2.1.4 MeetingRequest

The **MeetingRequest** type is an optional **container** type.

If a **MeetingRequest** type is defined, then it MUST contain one or more instances the **MeetingRequest.Recurrences** or **MeetingRequest.Categories** types.

2.2.1.5 MeetingRequest.Recurrences

The **MeetingRequest.Recurrences** type is a **container** type that contains a collection of **Recurrence** elements.

The **MeetingRequest.Recurrences** type is an optional child type of the **MeetingRequest** type.

2.2.1.6 MeetingRequest.Recurrences.Recurrence

The **MeetingRequest.Recurrences.Recurrence** type is a **container** type that describes when and how often this meeting request recurs.

If a **MeetingRequest.Recurrences** type is defined, then it MUST contain one or more instances of this type.

2.2.1.7 Flag

The **Flag** type is an optional container type that defines the flag associated with this item, along with the items current status.

The **Flag** type either contains one or more instances the **Flag** elements specified in [2.2.2.41](#) through [2.2.2.53](#), or if no flags are present, it SHOULD be included as an empty container, <Flag />.

2.2.1.8 Categories

The **Categories** type <2>is an optional container type that contains the user-selected categories for this message. The **Categories** type is defined as a complex type in the Email namespace.

An empty **Categories** type is included in a **Sync Add** command if no child **Categories.Category** elements have been set on the message. A **Sync Change** element containing an empty **Categories** type indicates that all the **Categories.Category** elements associated with the message have been removed. Not receiving a **Categories** type in a **Sync Change** message indicates that the **Categories.Category** value has not changed.

2.2.2 Elements

The following tables summarize the set of common XML schema element definitions defined by this specification. XML schema element definitions that are specific to a particular operation are described with that operation.

The E-mail class elements listed in the following table do not have any child elements in the command request or response. If child elements are included in a Sync request, the server returns a <Status> value of 6 in the **Sync** response.

Element	Description
To	The list of recipients .
Cc	The list of secondary recipients.
From	The e-mail address of the individual who sent the message.
Subject	The subject of the e-mail message.
ReplyTo	The e-mail address to which replies will be addressed by default.
DateReceived	The date and time that the message was received on the server.
DisplayTo	The names of the primary recipients of the message.
ThreadTopic	The topic used in conversation reading.
Importance	The importance of the message, as determined by the sender.
Read	Specifies whether the message has been read.
Attachments.Attachment.DisplayName	The name of the attachment file as displayed to the user.
Attachments.Attachment.UmAttOrder	The order of electronic voice mail attachments.
Attachments.Attachment.UmAttDuration	The duration of electronic voicemail attachments.
MessageClass	The message class of this e-mail message.
MeetingRequest.AllDayEvent	Indicates whether the calendar item is an all day event.
MeetingRequest.StartTime	The date and time that the MeetingRequest element starts.
MeetingRequest.DtStamp	The date and time that the calendar item was created.
MeetingRequest.EndTime	The date and time that the MeetingRequest element ends.

Element	Description
MeetingRequest.InstanceType	The type of calendar item.
MeetingRequest.Location	The location for the calendar item.
MeetingRequest.Organizer	The e-mail alias of the meeting organizer.
MeetingRequest.RecurrenceId	A specific instance of a recurring calendar item.
MeetingRequest.Reminder	The number of seconds prior to the calendar item's start time that a reminder is displayed.
MeetingRequest.ResponseRequested	Indicates whether the originator of the meeting has requested a response.
MeetingRequest.Recurrences.Recurrence.Type	The recurrence type of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.Interval	The interval between recurrences of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.Until	The end time of a series of recurrence meetings.
MeetingRequest.Recurrences.Recurrence.Occurrences	The number of occurrences before the series of recurring meeting ends.
MeetingRequest.Recurrences.Recurrence.WeekOfMonth	The week of the month of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.DayOfMonth	The day of the month of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.DayOfWeek	The day of the week of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.MonthOfYear	The month of the year of the recurring meeting.
MeetingRequest.Recurrences.Recurrence.CalendarType	The type of calendar associated with the recurrence.
MeetingRequest.Recurrences.Recurrence.IsLeapMonth	Specifies whether the recurrence takes place in the leap month of the given year.
MeetingRequest.Sensitivity	The confidentiality level of the meeting request.
MeetingRequest.BusyStatus	The intended Busy status for the meeting request.
MeetingRequest.TimeZone	The time zone specified when the calendar item was created.
MeetingRequest.GlobalObjId	A random 76-digit hexadecimal ID generated by the client for the meeting request.
MeetingRequest.DisallowNewTimeProposal	Indicates whether recipients can propose a new meeting time.
InternetCPID	The original code page ID from the MIME message.
Flag.Subject	The subject of the flag as it would appear in a task list.

Element	Description
Flag.Status	The current status of the flag.
Flag.FlagType	The value of the Flag To: follow up field.
Flag.DateCompleted	The date on which the flagged item was completed.
Flag.CompleteTime	The time at which the flagged item was marked as finished.
Flag.StartDate	The start date of the flagged item.
Flag.DueDate	The due date of the flagged item.
Flag.UTCStartDate	The Coordinated Universal Time (UTC) value of the local StartDate .
Flag.UTCDueEndDate	The UTC value of the local DueDate .
Flag.ReminderSet	Identifies whether a reminder has been set for this flagged item.
Flag.ReminderTime	The date and time that the reminder is scheduled to occur.
Flag.OrdinalDate	The time at which the client set the flag.
Flag.SubOrdinalDate	A string used to sort items.
NativeBodyType	The format in which the item is stored on the server.
ContentClass	The content class of the data.
UmCallerID	The callback telephone number of the person who called or left an electronic voice message.
UmUserNotes	User notes related to an electronic voice message.
ConversationId	A unique identifier for a conversation.
ConversationIndex	A ConversationId and a set of dates and times used by clients to generate a conversation tree view.
LastVerbExecuted	The last action, such as reply or forward, which was taken on the message so that the client can display the appropriate icon.
LastVerbExecutionTime	The time when the LastVerbExecuted was performed on the message.
ReceivedAsBcc	Indicates whether the recipient was blind carbon copied on a message.
Sender	The user that actually sent the message when the message was not sent by the user identified

Element	Description
	by From .
Categories.Category	The category for this e-mail item.

2.2.2.1 To

The **To** element is an optional element that specifies the list of primary recipients. It is defined as an element in the Email namespace.

The value of this element contains one or more e-mail addresses. If there are multiple e-mail addresses, then they are separated by commas.

The **To** element has a maximum length of 32,768 characters.

2.2.2.2 Cc

The **Cc** element is an optional element that specifies the list of secondary recipients of this message. It is defined as an element in the Email namespace.

The message is directed at the primary recipient as specified by the **To** element, but the secondary recipients also receive a copy of the message.

The value of this element contains one or more e-mail addresses. If there are multiple e-mail addresses, then they are separated by commas.

2.2.2.3 From

The **From** element is an optional element that specifies the e-mail address of the individual who sent this message. It is defined as an element in the Email namespace.

The value of this element contains one or more e-mail addresses. If there are multiple e-mail addresses, then they are separated by commas.

The **From** element has a maximum length of 32,768 characters.

2.2.2.4 Subject

The **Subject** element is an optional element that specifies the subject of the e-mail message. It is defined as an element in the Email namespace.

2.2.2.5 ReplyTo

The **ReplyTo** element is an optional element that specifies the e-mail address to which replies will be addressed by default. It is defined as an element in the Email namespace.

The value of this element contains one or more e-mail addresses. If there are multiple e-mail addresses, then they are separated by a semi-colon.

2.2.2.6 DateReceived

The **DateReceived** element is an optional element that specifies the date and time when this message was received by the current recipient. It is defined as an element in the Email namespace. The value of this element is a date/time value, as specified in [\[MS-ASDTYPE\]](#).

2.2.2.7 DisplayTo

The **DisplayTo** element is an optional element that specifies the e-mail addresses of the primary recipients of this message. It is defined as an element in the Email namespace.

The value of this element contains one or more display names. If there are multiple display names, then they are separated by semi-colons.

2.2.2.8 ThreadTopic

The **ThreadTopic** element is an optional element that specifies the topic used for conversation threading. It is defined as an element in the Email namespace.

2.2.2.9 Importance

The **Importance** element is an optional element that specifies the importance of the message, as determined by the sender. It is defined as an element in the Email namespace.

The value of this element MUST be one of the following.

Value	Meaning
0	Low importance
1	Normal importance
2	High importance

If this element is omitted, then clients MUST assume 1 as the default.

2.2.2.10 Read

The **Read** element is an optional element that specifies whether the e-mail message has been viewed by the current recipient. It is defined as an element in the Email namespace.

A value of **TRUE** indicates the e-mail message was viewed; a value of **FALSE** indicates the e-mail message was not viewed.

The value of this element is a **boolean** value, as specified in [\[MS-ASDTYPE\]](#).

2.2.2.11 Attachments.Attachment.DisplayName

The **Attachments.Attachment.DisplayName** element is an optional child element of the attachment type that specifies the name of the attachment file as displayed to the user. It is defined as an element in the Email namespace.

The **DisplayName** element is further specified in [\[MS-ASAIRS\]](#) section 2.2.2.5.

2.2.2.12 Attachments.Attachment.UmAttOrder

The **UmAttOrder** element [<3>](#) identifies the order of electronic voice mail attachments. It is defined as an element in the Email2 namespace.

This value is set by the server and is read-only for the client.

The most recent voice mail attachment in an e-mail item MUST have a **UmAttOrder** value of 1. Whenever a new electronic voice message associated with the same e-mail item is received, the new voice attachment is appended to the end of the list and all electronic voice attachments are renumbered.

This element MUST only be included on messages with a **MessageClass** prefix of **IPM.Note.Microsoft.Voicemail**, **IPM.Note.RPMSG.Microsoft.Voicemail**, or **IPM.Note.Microsoft.Missed.Voice**.

2.2.2.13 Attachments.Attachment.UmAttDuration

The **UmAttDuration** element [<4>](#) specifies the duration of the most recent electronic voice mail attachment in seconds. It is defined as an element in the Email2 namespace.

This element MUST only be used for electronic voice message attachments. This value is set by the server and is read-only for the client.

This element MUST only be included on messages with a **MessageClass** prefix of **IPM.Note.Microsoft.Voicemail**, **IPM.Note.RPMSG.Microsoft.Voicemail**, or **IPM.Note.Microsoft.Missed.Voice**.

2.2.2.14 MessageClass

The **MessageClass** element is an optional element that specifies the message class of this e-mail message. It is defined as an element in the Email namespace.

The **MessageClass** value is provided as a hint to the client to aid in processing the item. The protocol does not validate that the item has the correct **MessageClass** value, nor does it update incorrect values.

The value of the **MessageClass** element SHOULD be one of the following values. messages with values not contained in the following table will be opened as normal e-mail messages.

Value	Meaning
IPM.Note	Normal e-mail message
IPM.Note.SMIME	The message is encrypted and can also be signed.
IPM.Note.SMIME.MultipartSigned	The message is clear signed.
IPM.Note.Receipt.SMIME	The message is a secure read receipt .
IPM.InfoPathForm	An InfoPath form, as specified by [MS-IPFFX]
IPM.Schedule.Meeting.Request	Meeting request
IPM.Notification.Meeting	Meeting notification
IPM.Post	Post
IPM.Octel.Voice	Octel voice message
IPM.Voicenotes	Electronic voice notes
IPM.Sharing	Shared message

In addition, certain administrative messages have message classes that are derived from the message classes in the preceding table. The format is a prefix of REPORT and a suffix that indicates the type of report. For those administrative messages, the value of the **MessageClass** element MUST be one of the following values.

MessageClass	Description
REPORT.IPM.NOTE.NDR	Non-delivery report (NDR) for a standard message.
REPORT.IPM.NOTE.DR	Delivery receipt for a standard message.
REPORT.IPM.NOTE.DELAYED	Delivery receipt for a delayed message.
*REPORT.IPM.NOTE.IPNRN	Read receipt for a standard message.
*REPORT.IPM.NOTE.IPNNRN	Non-read receipt for a standard message.
REPORT.IPM.SCHEDULE.MEETING.REQUEST.NDR	NDR for a meeting request.
REPORT.IPM.SCHEDULE.MEETING.RESP.POS.NDR	NDR for a positive meeting response (accept).
REPORT.IPM.SCHEDULE.MEETING.RESP.TENT.NDR	NDR for a Tentative meeting response.
REPORT.IPM.SCHEDULE.MEETING.CANCELED.NDR	NDR for a cancelled meeting notification.
REPORT.IPM.NOTE.SMIME.NDR	NDR for a Secure MIME (S/MIME) encrypted and opaque-signed message.
*REPORT.IPM.NOTE.SMIME.DR	Delivery receipt for an S/MIME encrypted and opaque-signed message.
*REPORT.IPM.NOTE.SMIME.MULTIPARTSIGNED.NDR	NDR for an S/MIME clear signed message.
*REPORT.IPM.NOTE.SMIME.MULTIPARTSIGNED.DR	Delivery receipt for an S/MIME clear signed message.

2.2.2.15 MeetingRequest.AllDayEvent

The **MeetingRequest.AllDayEvent** element is a required child element of the **MeetingRequest** type that specifies whether this meeting request lasts the entire day. It is defined as an element in the Email namespace.

If the value of this element is set to 1, then the attached meeting request is an all day event. If the value of this element is set to 0, then the attached meeting request is not an all day event.

The value of this element is an **unsignedByte** value, as specified in [\[MS-ASDTYPE\]](#) section 2.10.

2.2.2.16 MeetingRequest.StartTime

The **MeetingRequest.StartTime** element is a required child element of the **MeetingRequest** type that specifies when this meeting begins. It is defined as an element in the Email namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

2.2.2.17 MeetingRequest.DtStamp

The **MeetingRequest.DtStamp** element is a required child element of the **MeetingRequest** type that specifies the date and time this calendar item was created. It is defined as an element in the Email namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

2.2.2.18 MeetingRequest.EndTime

The **MeetingRequest.EndTime** element is a required child element of the **MeetingRequest** type that specifies the date and time when the meeting ends. It is defined as an element in the Email namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

2.2.2.19 MeetingRequest.InstanceType

The **MeetingRequest.InstanceType** element is a required child element of the **MeetingRequest** type that specifies whether this is a single or recurring appointment. It is defined as an element in the Email namespace.

The value of this element is an enumeration which **MUST** be one of the following values:

Meaning	Meaning
0	Single appointment.
1	Master recurring appointment.
2	Single instance of a recurring appointment.
3	Exception to a recurring appointment.

2.2.2.20 MeetingRequest.Location

The **MeetingRequest.Location** element is an optional child element of the **MeetingRequest** type that specifies where this meeting will be held. This element has a maximum string length of 32,768 characters. It is defined as an element in the Email namespace. [<5>](#)

2.2.2.21 MeetingRequest.Organizer

The **MeetingRequest.Organizer** element is an optional child element of the **MeetingRequest** type that specifies who organized this meeting. It is defined as an element in the Email namespace.

The value of this element is an **e-mail address** as specified in [\[MS-ASDTYPE\]](#) section 2.5.

2.2.2.22 MeetingRequest.RecurrenceId

The **MeetingRequest.RecurrenceId** element is an optional child element of the **MeetingRequest** type that specifies the date and time of this recurrence of a recurring meeting. It is defined as an element in the Email namespace.

The server **MUST** include this element in response messages to indicate a single instance exception to a recurring meeting; otherwise, the server **MUST NOT** include this element.

The value of this element MUST be the date and time corresponding to this instance of a recurring item.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

2.2.2.23 MeetingRequest.Reminder

The **MeetingRequest.Reminder** element is an optional child element of the **MeetingRequest** type that specifies the number of seconds prior to the calendar item's start time that a reminder will be displayed. It is defined as an element in the Email namespace.

2.2.2.24 MeetingRequest.ResponseRequested

The **MeetingRequest.ResponseRequested** element is an optional child element of the **MeetingRequest** type that specified whether the organizer has requested a response to this meeting request. It is defined as an element in the Email namespace.

2.2.2.25 MeetingRequest.Recurrences.Recurrence.Type

The **MeetingRequest.Recurrences.Recurrence.Type** element is a required child element of the **Recurrence** type that specifies how this meeting recurs. It is defined as an element in the Email namespace.

The value of this element MUST be one of the following:

Value	Meaning
0	Recurs daily.
1	Recurs weekly.
2	Recurs monthly on the Nth day of the month.
3	Recurs monthly.
5	Recurs yearly on the Nth day of the Nth month each year.
6	Recurs yearly on the Nth day of the week of the Nth month each year.

2.2.2.26 MeetingRequest.Recurrences.Recurrence.Interval

The **MeetingRequest.Recurrences.Recurrence.Interval** element is a required child element of the **Recurrence** type that specifies the interval between recurrences. It is defined as an element in the Email namespace.

An **Interval** value of 1 indicates that the meeting occurs every week, month, or year, depending upon the value of **MeetingRequest.Recurrences.Recurrence.Type**. An **Interval** value of 2 indicates that the meeting occurs every other week, month, or year.

For examples showing how to set **MeetingRequest** recurrence intervals, see section [4.2](#).

2.2.2.27 MeetingRequest.Recurrences.Recurrence.Until

The **MeetingRequest.Recurrences.Recurrence.Until** element is a required child element of the **Recurrence** type that specifies the end date and time of a recurring meeting. It is defined as an element in the Email namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

2.2.2.28 MeetingRequest.Recurrences.Recurrence.Occurrences

The **MeetingRequest.Recurrences.Recurrence.Occurrences** element is a required child element of the **Recurrence** type that specifies the number of occurrences before the series of recurring meetings ends. It is defined as an element in the Email namespace.

2.2.2.29 MeetingRequest.Recurrences.Recurrence.WeekOfMonth

The **MeetingRequest.Recurrences.Recurrence.WeekOfMonth** element is an optional child element of the **Recurrence** type that specifies the week of the month in which this meeting recurs. It is defined as an element in the Email namespace.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 5 (recurs yearly on the Nth day of the Nth month each year).

2.2.2.30 MeetingRequest.Recurrences.Recurrence.DayOfMonth

The **MeetingRequest.Recurrences.Recurrence.DayOfMonth** element is an optional child element of the **Recurrence** type that specifies the day of the month on which this meeting recurs. It is defined as an element in the Email namespace.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 3 (recurs monthly) or 6 (recurs yearly on the Nth day of the week of the Nth month each year).

For examples showing how to set **MeetingRequest** recurrence intervals, see section [4.2](#).

2.2.2.31 MeetingRequest.Recurrences.Recurrence.DayOfWeek

The **MeetingRequest.Recurrences.Recurrence.DayOfWeek** element is an optional child element of the **Recurrence** type that specifies the day of the week on which this meeting recurs. It is defined as an element in the Email namespace.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 1 (recurs weekly), 2 (recurs monthly on the Nth day of the month), or 6 (recurs yearly on the Nth day of the week of the Nth month each year).

The value of this element **MUST** be the sum of a minimum of one and a maximum of seven independent values from the following table.

Value	Meaning
1	Sunday
2	Monday
4	Tuesday
8	Wednesday
16	Thursday
32	Friday

Value	Meaning
64	Saturday

These values can be added together to specify that the meeting occurs on more than one day of the week.

For examples showing how to set different **MeetingRequest** recurrence intervals, see section [4.2](#).

2.2.2.32 MeetingRequest.Recurrences.Recurrence.MonthOfYear

The **MeetingRequest.Recurrences.Recurrence.MonthOfYear** element is an optional child element of the **Recurrence** type that specifies the month of the year in which this meeting recurs. It is defined as an element in the Email namespace.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 6, indicating that the meeting recurs yearly on the nth day of the week of the nth month.

2.2.2.33 MeetingRequest.Recurrences.Recurrence.CalendarType

The **CalendarType**[<6>](#) element is an optional child element of the **Recurrence** type that specifies the type of calendar associated with the recurrence. This element is required in server responses and is optional in client requests. A default value of "0" is used if the value is not specified. It is defined as an element in the Email2 namespace.

The following table lists valid values for the CalendarType element.

Value	Meaning
0	Default
1	Gregorian
2	Gregorian US
3	Japan
4	Taiwan
5	Korea
6	Hijri
7	Thai
8	Hebrew
9	GregorianMeFrench
10	Gregorian Arabic
11	Gregorian translated English
12	Gregorian translated French
14	Japanese Lunar

Value	Meaning
15	Chinese Lunar
20	Korean Lunar

Clients that send an unsupported CalendarType value to the server will receive an error.

2.2.2.34 MeetingRequest.Recurrences.Recurrence.IsLeapMonth

The **IsLeapMonth** [<7>](#) element is an optional child element of the **Recurrence** type that specifies whether the recurrence takes place in the leap month of the given year. It is defined as an element in the Email2 namespace.

This element is required in server responses and is optional in client requests. A default value of FALSE (0) is used if the value is not specified.

2.2.2.35 MeetingRequest.Sensitivity

The **MeetingRequest.Sensitivity** element is an optional child element of the **MeetingRequest** type that specifies the confidentiality level of the meeting request. It is defined as an element in the Email namespace.

The value of this element MUST be one of the following values.

Value	Meaning
0	Normal
1	Personal
2	Private
3	Confidential

If this element is missing, then a default of 0 MUST be assumed.

2.2.2.36 MeetingRequest.BusyStatus

The **MeetingRequest.BusyStatus** element is an optional child element of the **MeetingRequest** type that specifies whether the recipient of this meeting request is Busy at the specified time. It is defined as an element in the Email namespace.

The value of this element MUST be one of the following values.

Value	Meaning
0	Busy
1	Free
2	Tentative
3	Out of Office (OOF)

If this element is missing, then a default of 1 MUST be assumed.

2.2.2.37 MeetingRequest.TimeZone

The **MeetingRequest.TimeZone** element is a required child element of the **MeetingRequest** type that specifies the time zone specified when the calendar item was created. It is defined as an element in the Email namespace.

The value of this element is a **String** with a TimeZone format, as specified in [\[MS-ASDTYPE\]](#) section 2.7.

2.2.2.38 MeetingRequest.GlobalObjId

The **MeetingRequest.GlobalObjId** element is a required child element of the **MeetingRequest** type that contains a random 76-digit hexadecimal ID generated by the client for the meeting request. It is defined as an element in the Email namespace.

2.2.2.39 MeetingRequest.DisallowNewTimeProposal

The **DisallowNewTimeProposal** element [<8>](#) is an optional element that indicates whether recipients can propose a new meeting time. If the value is not specified, the value defaults to zero (0), meaning that new time proposals are allowed. A non-zero value indicates that new time proposals are not allowed. **DisallowNewTimeProposal** is defined as an element in the Email namespace.

The value of this element is an **unsignedByte** value, as specified in [\[MS-ASDTYPE\]](#) section 2.10.

2.2.2.40 InternetCPID

The **InternetCPID** element is a required element that contains the original code page ID from the MIME message. It is defined as an element in the Email namespace.

2.2.2.41 Flag.Subject

The **Flag.Subject** element is an optional child element of the **Flag** type that specifies the subject of the flag. It is defined as an element in the Tasks namespace.

The client or server SHOULD set the value of this element to the subject of the message when an item is flagged.

A maximum of one **Flag.Subject** element is allowed per **Flag**.

2.2.2.42 Flag.Status

The **Flag.Status** element is an optional child element of the **Flag** type that specifies the current status of the flag. It is defined as an element in the Email namespace.

The value of this element MUST be one of the following.

Value	Meaning
Null	Clear the flag
0	Clear the flag
1	Status is set to complete
2	Status is set to active

The **Sync** command response includes a **Status** value of 6 ([\[MS-ASCMD\]](#) section 2.2.1.19.2.16) if **Flag.Status** is set to a value other than 0, 1 or 2.

A maximum of one **Flag.Status** element is allowed per **Flag**.

2.2.2.43 Flag.FlagType

The **Flag.FlagType** element is an optional child element of the **Flag** type that specifies the flag type. It is defined as an element in the Email namespace.

Flag.FlagType is not required if the e-mail message is a meeting request or response.

This value is customizable, and is commonly set to "Flag for follow up" or "for Follow Up".

A maximum of one **Flag.FlagType** element is allowed per **Flag**.

2.2.2.44 Flag.DateCompleted

The **Flag.DateCompleted** element is an optional child element of the **Flag** type that identifies the date on which a flagged item was completed. It is defined as an element in the Tasks namespace.

The **DateCompleted** element is required to mark a flagged item as complete.

If the message includes a value for **Flag.DateCompleted**, then **Flag.CompleteTime** is also required.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.DateCompleted** element is allowed per **Flag**.

2.2.2.45 Flag.CompleteTime

The **Flag.CompleteTime** element is an optional element of the **Flag** type that identifies the time at which a flagged item was marked as finished. It is defined as an element in the Email namespace.

The **CompleteTime** element is required to mark a flagged item as complete.

If the message includes a value for **Flag.CompleteTime**, then **Flag.DateCompleted** is also required. The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.CompleteTime** element is allowed per **Flag**.

2.2.2.46 Flag.StartDate

The <Flag.StartDate> element is an optional child element of the <Flag> type that specifies when this flagged item was begun. It is defined as an element in the Tasks namespace.

When a flag is being updated, <Flag.StartDate> MUST NOT occur after <Flag.DueDate>. <Status> value 6 is returned in the **Sync** response if this condition is not met.

To set a flag, <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> MUST all be set, or MUST all be NULL. <Status> value 6 is returned in the **Sync** response if this condition is not met.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one <Flag.StartDate> element is allowed per <Flag>.

2.2.2.47 Flag.DueDate

The <Flag.DueDate> element is an optional child element of the <Flag> type that specifies when this flagged item is due. It is defined as an element in the Tasks namespace.

When a flag is being updated, <Flag.DueDate> MUST NOT occur before <Flag.StartDate>. <Status> value 6 is returned in the **Sync** response if this condition is not met.

To set a flag, <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> MUST all be set, or MUST all be NULL. <Status> value 6 is returned in the **Sync** response if this condition is not met.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one <Flag.DueDate> element is allowed per <Flag>.

2.2.2.48 Flag.UTCStartDate

The <Flag.UTCStartDate> element is an optional child element of the <Flag> type that contains the UTC value of the local <Flag.StartDate>. It is defined as an element in the Tasks namespace.

When a flag is being updated, <Flag.UTCStartDate> MUST occur before <Flag.UTCDueDate>. <Status> value 6 is returned in the **Sync** response if this condition is not met.

To set a flag, <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> MUST all be set, or MUST all be NULL. <Status> value 6 is returned in the **Sync** response if this condition is not met.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.UTCStartDate** element is allowed per **Flag**.

2.2.2.49 Flag.UTCDueDate

The <Flag.UTCDueDate> element is an optional child element of the <Flag> type that contains the UTC value of local <Flag.DueDate>. It is defined as an element in the Tasks namespace.

When a flag is being updated, <Flag.UTCDueDate> MUST NOT occur before <Flag.UTCStartDate>. <Status> value 6 is returned in the **Sync** response if this condition is not met.

To set a flag, <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> MUST all be set, or MUST all be NULL. <Status> value 6 is returned in the **Sync** response if this condition is not met.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.UTCDueDate** element is allowed per **Flag**.

2.2.2.50 Flag.ReminderSet

The **Flag.ReminderSet** element is an optional child element of the **Flag** type that is 1 if a reminder has been set for this task; otherwise it is set to zero (0). The default value is zero (0). It is defined as an element in the Tasks namespace.

The value of this element is an **unsignedByte** value, as specified in [\[MS-ASDTYPE\]](#) section 2.10.

A maximum of one **Flag.ReminderSet** element is allowed per **Flag**.

2.2.2.51 Flag.ReminderTime

The **Flag.ReminderTime** element is an optional child element of the **Flag** type that identifies the date and time that the reminder is scheduled to occur. It is defined as an element in the Tasks namespace.

<Flag.ReminderTime> MUST be set if <Flag.ReminderSet> is **TRUE** (1). A <Status> value of 6 is returned by the server in the **Sync** response if <Flag.ReminderSet> is TRUE (1) and <Flag.ReminderTime> is not included in a the **Sync** request.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.ReminderTime** element is allowed per **Flag**.

2.2.2.52 Flag.OrdinalDate

The **Flag.OrdinalDate** element is an optional child element of the **Flag** type that identifies the time at which the client set the flag. It is defined as an element in the Tasks namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.OrdinalDate** element is allowed per **Flag**.

2.2.2.53 Flag.SubOrdinalDate

The **Flag.SubOrdinalDate** element is an optional child element of the **Flag** type that is used for sorting. The value can be any string and can be used for additional sorting if there are duplicate OrdinalDates. It is defined as an element in the Tasks namespace.

The value of this element is a **date/time** value, as specified in [\[MS-ASDTYPE\]](#) section 2.6.

A maximum of one **Flag.SubOrdinalDate** element is allowed per **Flag**.

2.2.2.54 NativeBodyType

The **NativeBodyType** element is an optional element that specifies how the e-mail message is stored on the server.

For details about the **NativeBodyType** element, see [\[MS-ASAIRS\]](#) section 2.2.2.10.

2.2.2.55 ContentClass

The **ContentClass** element is an optional element that specifies the content class of the data. For e-mail messages, the value of this element MUST be set to "urn:content-classes:message". It is defined as an element in the Email namespace.

2.2.2.56 UmCallerID

The **UmCallerID** element [<9>](#) is an optional element that specifies the callback telephone number of the person who called or left an electronic voice message. It is defined as an element in the Email2 namespace.

This property is sent from the server to the client, and MUST NOT be sent from the client to the server. <Status> value 6 is returned in a **Sync** command response if the client attempts to send the <UmCallerID> element to the server. The **UmCallerID** element is not included, or is empty, if the call originated as a private, blocked, or otherwise anonymous call. The **string** is either formatted as

an E.164 telephone number (as specified in [\[E164\]](#)) or a session initiated protocol link to initiate a **Voice over Internet Protocol (VoIP)** call. For more details about session initiated protocol links, see [\[RFC3261\]](#) section 19.1.

This element MUST only be included on messages with one of the following **MessageClass** values:

- IPM.Note.Microsoft.Voiceemail
- IPM.Note.Microsoft.Voiceemail.UM
- IPM.Note.Microsoft.Voiceemail.UM.CA
- IPM.Note.RPMSG.Microsoft.Voiceemail
- IPM.Note.RPMSG.Microsoft.Voiceemail.UM
- IPM.Note.RPMSG.Microsoft.Voiceemail.UM.CA
- IPM.Note.Microsoft.Missed.Voice

Only one **UmCallerID** element is allowed per message. In order to enable future VoIP scenarios, the server SHOULD send this field to clients regardless of the client's current VoIP capabilities.

2.2.2.57 UmUserNotes

The **UmUserNotes** element [<10>](#) is an optional property that contains user notes related to an electronic voice message. It is defined as an element in the Email2 namespace.

This property is sent from the server to the client, and MUST NOT be sent from the client to the server. <Status> value 6 is returned in a **Sync** command response if the client attempts to send the <UmUserNotes> element to the server.

This element MUST only be included on messages with one of the following **MessageClass** values:

- IPM.Note.Microsoft.Voiceemail
- IPM.Note.Microsoft.Voiceemail.UM
- IPM.Note.Microsoft.Voiceemail.UM.CA
- IPM.Note.RPMSG.Microsoft.Voiceemail
- IPM.Note.RPMSG.Microsoft.Voiceemail.UM
- IPM.Note.RPMSG.Microsoft.Voiceemail.UM.CA
- IPM.Note.Microsoft.Missed.Voice

Only one **UserNotes** element is allowed for any message. The server truncates notes larger than 32K, to 32K.

2.2.2.58 ConversationId

The **ConversationId** element [<11>](#) is a required byte-array that specifies a unique identifier for a conversation. It is defined as an element in the Email2 namespace.

The client MUST NOT change the **ConversationId** value. <Status> value 6 is returned in the **Sync** response command when a <Change> is attempted on the <ConversationId>.

The <ConversationId> content is transferred as an opaque **BLOB** within the WBXML tags.

For more details about conversations, see [\[MS-ASCON\]](#).

2.2.2.59 ConversationIndex

The **ConversationIndex** element [<12>](#) is a required byte-array that contains a set of timestamps used by clients to generate a conversation tree view. The first timestamp identifies the date and time when the message was originally sent by the server. Additional timestamps are added when the message is forwarded or replied to. The **ConversationIndex** element is defined as an element in the Email2 namespace.

The first 5 byte timestamp in the **ConversationIndex** identifies the system time when the message was sent by the server, converted to the FILETIME structure format, as specified in [MS-DTYP] section 2.3.1.

Each additional timestamp contains 5 bytes, divided as follows:

- One bit containing a code representing the difference between the current time and the time stored in the first timestamp. This bit is 0 if the difference is less than .02 seconds and greater than two years and 1 if the difference is less than one second and greater than 56 years.
- Thirty one bits containing the difference between the current time and the time in the first timestamp, expressed in FILETIME units. This part of the value is produced using one of two strategies, depending on the value of the first bit. If this bit is zero, the server discards the high 15 bits and the low 18 bits. If this bit is one, the server discards the high 10 bits and the low 23 bits.
- Four bits containing a random number generated by calling the Win32 function GetTickCount.
- Four bits containing a sequence count that is taken from part of the random number.

The <ConversationIndex> content is transferred as an opaque BLOB within the WBXML tags.

The client MUST NOT change the **ConversationIndex** value. <Status> value 6 is returned in the **Sync** response command when a <Change> is attempted on the <ConversationIndex>.

2.2.2.60 LastVerbExecuted

The **LastVerbExecuted** element [<13>](#) is an optional element that indicates the last **action**, such as reply or forward, that was taken on the **message** so that the client can display the appropriate icon. It is defined as an element in the Email2 namespace.

The following **table** lists valid integer values for the element.

Value	Meaning
0	Unknown
1	REPLYTOSENDER
2	REPLYTOALL
3	FORWARD

2.2.2.61 LastVerbExecutionTime

The **LastVerbExecutionTime** element [<14>](#) is an optional **datetime** element that indicates the time when the **LastVerbExecuted** (section [2.2.2.60](#)) was performed on the message. It is defined as an element in the Email2 namespace.

2.2.2.62 ReceivedAsBcc

The **ReceivedAsBcc** element [<15>](#) is an optional **boolean** value that notifies the user that they were blind carbon copied on an e-mail. It is defined as an element in the Email2 namespace.

Clients MUST not change the **ReceivedAsBcc** value. If the client changes the value, the server sets the **Status** element of the **Sync** command response to "6", as specified in [\[MS-ASCMD\]](#) section 2.2.1.19.2.16.

The **ReceivedAsBcc** value is not sent if the value is FALSE (0).

2.2.2.63 Sender

The **Sender** element [<16>](#) is an optional value that indicates that the message was not sent from the user identified by the **From** element. It is defined as an element in the Email2 namespace.

This element is set by the server and is read-only on the client. If the client attempts to change this value, then the server sets the **Status** element of the **Sync** command response to "6", as specified in [\[MS-ASCMD\]](#) section 2.2.1.19.2.16.

If included, the **Sender** element identifies the user that actually sent the message, and the **From** element identifies the user on whose behalf the message was sent. Use of the **Sender** element indicates that the sender of the item had **delegate** access to the **From** user's **mailbox**.

The client displays the message as <Sender> on behalf of <From>.

The **Sender** element is not sent to the client when **Sender** and **From** have the same value, or when the **Sender** element is NULL.

2.2.2.64 Categories.Category

The **Categories.Category** element [<17>](#) is an optional element of the Categories type that specifies a category for this e-mail item.

A command request SHOULD limit itself to no more than 300 **Categories.Category** elements per Categories type.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

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

E-mail class: A set of complex types and elements that specifies an e-mail message. E-mail class data is included in command requests sent to the server when e-mail messages need to be retrieved or synchronized. For more details about processing command requests, see section [3.1.5](#).

Command request: A WBXML formatted message that adheres to the command schemas specified in [\[MS-ASCMD\]](#).

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Synchronize E-mail Between Client and Server

A client initiates **synchronization** of E-mail class data with the server by sending a **Sync** command request, as specified in section [3.1.5.3](#) and [\[MS-ASCMD\]](#) section 2.2.1.19.1.

3.1.4.2 Search E-mail

A client searches a **message database (MDB)** for E-mail class data by sending a **Search** command request to the server, as specified in section [3.1.5.2](#) and [\[MS-ASCMD\]](#) section 2.2.1.14.1.

3.1.4.3 Retrieve Individual E-mail

E-mail data for one or more individual e-mail items is requested by the client using an **ItemOperations** command request, which is a wrapper for the **Fetch** command. An **ItemOperations** command can contain multiple **Fetch** commands. The **ItemOperations** command request is specified in section [3.1.5.1](#) and [\[MS-ASCMD\]](#) section 2.2.1.8.2.

3.1.4.4 Send Flag Updates to the Server

Basic flagging enables clients to flag e-mail messages, mark flags as complete, or clear flags. Flags are specified in section [2.2.1.7](#), and section [2.2.2.41](#) through section [2.2.2.53](#).

The following figure shows the life cycle of a flag.

Figure 1: Flag life cycle



For every flag update that is sent from the client, the server can update the flag on the e-mail message using the **Change** element of the **Sync** command. The server uses the logic provided in the following table to determine which flag action (clear, set, mark complete) to invoke when updating flag status based on the value of the **Status** element, as specified in section [2.2.2.42](#).

Action	Required Properties from Device
Flag an item (basic)	Status = 2 FlagType = "Flag for follow up" StartDate and UTCStartDate DueDate and UTCDueDate or Status = 2 DateCompleted
Flag an item (task flagging)	Status = 2 Subject = <user defined> FlagType = "Flag for follow up" StartDate and UTCStartDate DueDate and UTCDueDate ReminderSet ReminderTime or Status = 2 DateCompleted
Mark an item complete (basic)	Status = 1 CompleteTime DateCompleted
Mark an item complete (task flagging)	Status = 1 CompleteTime DateCompleted
Clearing the flag on an item	Status = 0 or Flag node empty

Action	Required Properties from Device
Clearing the flag on an item (task flagging)	Status = 0 or Flag node empty
Update the flag metadata (basic)	All updated properties
Update flag metadata (task flagging)	All updated properties

The **Sync** command response includes a **Status** value of "6" ([\[MS-ASCMD\]](#) section 2.2.1.19.2.16) if any of the required elements listed in the table are missing from the **Sync** request.

FlagType is not required if the item is a meeting request or response message.

3.1.5 Message Processing Events and Sequencing Rules

The following sections define how various elements of the E-mail class are used in the context of specific commands. For more details about the commands themselves, see [\[MS-ASCMD\]](#).

3.1.5.1 ItemOperations Command Request

A client sends the **ItemOperations** command request to the server to retrieve specific E-mail class items by using the **Fetch** element. An **ItemOperations** request can contain multiple **Fetch** elements.

Only the top level schema properties in the E-mail class can be included in an **ItemOperations** command request. The top level schema properties in the Email class are:

- **To**
- **Cc**
- **From**
- **ReplyTo**
- **DateReceived**
- **Subject**
- **DisplayTo**
- **Importance**
- **Read**
- **MessageClass**
- **MeetingRequest**
- **ThreadTopic**
- **InternetCPID**

The top level schema properties MUST be transmitted as children of the **Schema** type ([\[MS-ASCMD\]](#) section 2.2.1.8.2.13).

The **ItemOperations** command request is further specified in [\[MS-ASCMD\]](#) section 2.2.1.8.2.

3.1.5.2 Search Command Request

A client sends a **Search** command request to the server to retrieve E-mail class items that match the criteria specified by the client.

The complex types and elements for the E-mail class MUST NOT be included in a **Search** command request. <Status> value of 2 is returned as a child of the <Store> element in a <Search> response if this condition is not met.

The **Search** command request is further specified in [\[MS-ASCMD\]](#) section 2.2.1.14.1.

3.1.5.3 Sync Command Request

A client sends a **Sync** command request to the server to synchronize its E-mail class items for a specific user with the E-mail items currently stored by the server.

E-mail class complex types and elements MUST be transmitted as children of the **ApplicationData** type, as specified in [\[MS-ASCMD\]](#) section 2.2.1.19.1.7.

The **ApplicationData** type MUST only contain the following Email class elements or types in a **Sync Change** request:

- **Flag**
- **Read**
- **Categories**

The **Sync** command request is further specified in [\[MS-ASCMD\]](#) section 2.2.1.19.1.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

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

E-mail class: A set of complex types and elements that specifies an e-mail message and adheres to the schema definition specified in section [2.2](#). E-mail class data is included in command responses sent to the client when e-mail messages have been retrieved, searched, or synchronized. For more details about processing command responses, see section [3.2.5](#).

Command response: A WBXML formatted message that adheres to the command schemas specified in [\[MS-ASCMD\]](#). The server can return zero or more E-mail class blocks in its command response, depending on how many e-mail items match the criteria specified in the client command request. One E-mail class **XML** block is returned by the server for every e-mail that matches the criteria specified in the client command request.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Synchronize E-mail Between Client and Server

Synchronization of E-mail class data is initiated by the client, as specified in section [3.1.4.1](#). The server responds with a **Sync** command response, as specified in [3.2.5.3](#) and [\[MS-ASCMD\]](#) section 2.2.1.19.2.

3.2.4.2 Search E-mail

Searching E-mail class data is initiated by the client, as specified in section [3.1.4.2](#). The server responds with a **Search** command response, as specified in section [3.2.5.2](#) and [\[MS-ASCMD\]](#) section 2.2.1.14.2.

3.2.4.3 Retrieve Individual E-mail

Retrieving E-mail data is initiated by the client, as specified in section [3.1.4.3](#). The server responds with an **ItemOperations** command response, as specified in section [3.2.5.1](#) and [\[MS-ASCMD\]](#) section 2.2.1.8.3.

3.2.4.4 Change Tracking Semantics for Flagging Properties

The **Flag** elements are tracked as a block. A server change to any **Flag** child element on the item MUST result in the whole **Flag** block being sent to the client in the **Change** element of the **Sync** command.

Implicit deletes are assumed when the change is applied based on the properties. An implicit delete means that if a tag is not found in the **Flag** container, then the property is deleted.

3.2.4.5 Send Changes to the Client

A server can partition e-mail changes into one or more of the following categories:

- Changes to the <Read> flag
- Changes to <Flag> elements
- Changes to other e-mail elements, such as <Subject>
- Changes to non-protocol properties

The server SHOULD send just the <Read> flag or <Flag> elements if that is all that changed. The server MUST send down the entire E-mail item if any elements other than the <Read> flag or <Flag> elements changed. The server SHOULD send nothing to the client if the non-protocol properties changed.

The following matrix specifies what the server MUST send to the client based on which of the previous categories characterize the change.

Read Flag	Flag Elements	Other ActiveSync Protocol Properties	Non-ActiveSync Protocol Properties	Action
N	N	N	N	Send nothing to client
N	N	N	Y	Send nothing to client
N	N	Y	N	Send full item <Change> to client
N	N	Y	Y	Send full item <Change> to client
N	Y	N	N	Send <Flag> block only
N	Y	N	Y	Send <Flag> block only
N	Y	Y	N	Send full item <Change> to client
N	Y	Y	Y	Send full item <Change> to client
Y	N	N	N	Send <Read> flag only
Y	N	N	Y	Send <Read> flag only
Y	N	Y	N	Send full item <Change> to client
Y	N	Y	Y	Send full item <Change> to client
Y	Y	N	N	Send <Read> flag and <Flag> block
Y	Y	N	Y	Send <Read> flag and <Flag> block
Y	Y	Y	N	Send full item <Change> to client

3.2.5 Message Processing Events and Sequencing Rules

The following sections define how various elements of the E-mail class are used in the context of specific commands. For more details about the commands themselves, see [\[MS-ASCMD\]](#).

3.2.5.1 ItemOperations Command Response

Any of the complex types and elements in the E-mail class can be included in an **ItemOperations** command response. If a **Schema** element was included in the command request, then the complex types returned MUST be restricted to the complex types included in the command request's **Schema** element.

E-mail class complex types MUST be returned as children of the properties type ([\[MS-ASCMD\]](#) section 2.2.1.8.3.10).

The **ItemOperations** command response is further specified in [\[MS-ASCMD\]](#) section 2.2.1.8.3.

3.2.5.2 Search Command Response

Any of the E-mail class complex types and elements can be included in a **Search** command response as children of the properties element, as specified in [\[MS-ASCMD\]](#) section 2.2.1.14.2.2.

The **Search** command response is further specified in [\[MS-ASCMD\]](#) section 2.2.1.14.2.

3.2.5.3 Sync Command Response

Any of the E-mail class complex types and elements can be included in a **Sync** command response.

E-mail class complex types and elements MUST be returned as children of the **ApplicationData** type, as specified in [\[MS-ASCMD\]](#) section 2.2.1.19.2.2.

The **Sync** command response is further specified in [\[MS-ASCMD\]](#) section 2.2.1.19.2.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

The examples in this section use decoded values of the **URI** query parameters and the message body for clarity. The URI query parameter is base64-encoded and the body is WBXML-encoded when sent across the wire. For more details about the base-64 encoding used in the URI query parameter, see [\[MS-ASHTTP\]](#) section 2.2.1.1.1. For more details about WBXML encoding, see [\[MS-ASWBXML\]](#).

4.1 Synchronizing E-Mail

4.1.1 Example Sync Request for Inbox with Metadata

The following example **Sync** command request synchronizes all metadata in the Inbox, as specified by the **CollectionId** element. This request does not specify that the response include the body content, only the e-mail metadata will be included in the response.

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>927479200</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
    </Collection>
  </Collections>
</Sync>
```

The response to this command is in section [4.1.2](#).

4.1.2 Example Sync Response for Inbox with Metadata

The following example shows the response to the **Sync** command request in section [4.1.1](#). In the response, the server includes the metadata for one e-mail item. The estimated size and body type is included in the response, but the body of the message itself is not included.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 29 Jul 2009 19:25:46 GMT
Content-Length: 392

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:A1="POOMCONTACTS:" xmlns:A2="POOMMAIL:" xmlns:A3="AirNotify:" xmlns:A4="POOMCAL:"
xmlns:A5="Move:" xmlns:A6="GetItemEstimate:" xmlns:A7="FolderHierarchy:"
xmlns:A8="MeetingResponse:" xmlns:A9="POOMTASKS:" xmlns:A10="ResolveRecipients:"
xmlns:A11="ValidateCert:" xmlns:A12="POOMCONTACTS2:" xmlns:A13="Ping:" xmlns:A14="Provision:"
xmlns:A15="Search:" xmlns:A16="Gal:" xmlns:A17="AirSyncBase:" xmlns:A18="Settings:"
xmlns:A19="DocumentLibrary:" xmlns:A20="ItemOperations:" xmlns:A21="ComposeMail:"
xmlns:A22="POOMMAIL2:" xmlns:A23="Notes:" xmlns="AirSync:">
```



```

<Collections>
  <Collection>
    <SyncKey>927479210</SyncKey>
    <CollectionId>5</CollectionId>
    <Status>1</Status>
    <Commands>
      <Add>
        <ServerId>5:1</ServerId>
        <ApplicationData>
          <A2:To>"Device User" &lt;deviceuser@example.com></A2:To>
          <A2:From>"Device User 2" &lt;deviceuser2@example.com></A2:From>
          <A2:Subject>New mail message</A2:Subject>
          <A2:DateReceived>2009-07-29T19:25:37.817Z</A2:DateReceived>
          <A2:DisplayTo>Device User</A2:DisplayTo>
          <A2:ThreadTopic>New mail message</A2:ThreadTopic>
          <A2:Importance>1</A2:Importance>
          <A2:Read>0</A2:Read>
          <A17:Body>
            <A17:Type>2</A17:Type>
            <A17:EstimatedDataSize>116575</A17:EstimatedDataSize>
            <A17:Truncated>1</A17:Truncated>
          </A17:Body>
          <A2:MessageClass>IPM.Note</A2:MessageClass>
          <A2:InternetCPIID>1252</A2:InternetCPIID>
          <A2:Flag />
          <A2:ContentClass>urn:content-classes:message</A2:ContentClass>
          <A17:NativeBodyType>2</A17:NativeBodyType>
          <A22:ConversationId>FF68022058BD485996BE15F6F6D99320</A22:ConversationId>
          <A22:ConversationIndex>CA2CFA8A23</A22:ConversationIndex>
          <A2:Categories />
        </ApplicationData>
      </Add>
    </Commands>
  </Collection>
</Collections>
</Sync>

```

4.1.3 Example Sync Request for Inbox with Body Preferences

The following example **Sync** command request includes the **BodyPreference** option, which specifies that the client wants HTML mail (Type 2) with the bodies truncated to 5,120 bytes (5 KB). Request messages can include multiple **BodyPreference** elements to specify different **TruncationSize** values for each **Type** value. For more information about the **BodyPreference**, **TruncationSize**, or **Type** elements, see [\[MS-ASAIRS\]](#) section 2.2.

```

MIME Support
POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:AirSyncBase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>

```

```

    <SyncKey>1534587728</SyncKey>
    <CollectionId>5</CollectionId>
    <DeletesAsMoves>1</DeletesAsMoves>
    <GetChanges>1</GetChanges>
    <WindowSize>512</WindowSize>
    <Options>
      <MIMESupport>0</MIMESupport>
      <AirSyncBase:BodyPreference>
        <AirSyncBase:Type>2</AirSyncBase:Type>
        <AirSyncBase:TruncationSize>5120</AirSyncBase:TruncationSize>
      </AirSyncBase:BodyPreference>
    </Options>
  </Collection>
</Collections>
</Sync>

```

The response to this message is in section [4.1.4](#).

4.1.4 Example Sync Response Adding One HTML Message

The following example shows the response to the **Sync** command request in section [4.1.3](#). In the response, the server includes the metadata and body for one HTML e-mail message.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 19 Feb 2009 01:43:34 GMT
Content-Length: 763

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:Email2="POOMMAIL2" xmlns:AirSyncBase="AirSyncBase:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1174511196</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>5:10</ServerId>
          <ApplicationData>
            <Email:To>"Device User" &lt;deviceuser@example.com&gt;</Email:To>
            <Email:From>"Device User2" &lt;deviceuser2@example.com&gt;</Email:From>
            <Email:Subject>Sample HTML message</Email:Subject>
            <Email:DateReceived>2009-02-19T01:43:25.266Z</Email:DateReceived>
            <Email:DisplayTo>Device User</Email:DisplayTo>
            <Email:ThreadTopic>Sample HTML message</Email:ThreadTopic>
            <Email:Importance>1</Email:Importance>
            <Email:Read>0</Email:Read>
            <AirSyncBase:Body>
              <AirSyncBase:Type>2</AirSyncBase:Type>
              <AirSyncBase:EstimatedDataSize>375</AirSyncBase:EstimatedDataSize>
              <A17:Data>&lt;html dir="ltr"&gt;
                &lt;head&gt;
                  &lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8"&gt;
                  &lt;style id="owaParaStyle"&gt;
                    &lt;!--
                    p

```

```

        {margin-top:0px;
        margin-bottom:0px}
        --&gt;
        &lt;/style&gt;
        &lt;/head&gt;
        &lt;body&gt;
        &lt;div style="font-size:13px; color:#000000; direction:ltr; font-
family:Tahoma"&gt;
        &lt;div&gt;This is&nbsp;the body of an&nbsp;HTML e-mail
message.&lt;/div&gt;
        &lt;/div&gt;
        &lt;/body&gt;
        &lt;/html&gt;
        </A17:Data>
    </AirSyncBase:Body>
    <Email:MessageClass>IPM.Note</Email:MessageClass>
    <Email:InternetCPID>28591</Email:InternetCPID>
    <Email:Flag />
    <Email:ContentClass>urn:content-classes:message</Email:ContentClass>
    <AirSyncBase:NativeBodyType>2</AirSyncBase:NativeBodyType>
    <Email2:ConversationId>FF68022058BD485996BE15F6F6D99320</Email2:ConversationId>
    <Email2:ConversationIndex>CA2CFA8A23</Email2:ConversationIndex>
    <Email:Categories />
    </ApplicationData>
</Add>
<Change>
    <ServerId>5:8</ServerId>
    <ApplicationData>
        <Email:Read>1</Email:Read>
    </ApplicationData>
</Change>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.5 Example Sync Response Adding an Electronic Voice Mail Attachment

The following example shows the **Sync** command response adding one e-mail item with an electronic voice mail **attachment** to the client.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 04 Feb 2009 23:50:03 GMT
Content-Length: 378

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:Email="POOMMAIL:" xmlns:AirSyncBase="AirSyncBase:"
xmlns:Email2="POOMMAIL2:">
    <Collections>
        <Collection>
            <SyncKey>1336143213</SyncKey>
            <CollectionId>20</CollectionId>
            <Status>1</Status>
            <Commands>
                <Add>
                    <ServerId>20:2</ServerId>

```

```

<ApplicationData>
  <Email:To>"Device User" <deviceuser@example.com></Email:To>
  <Email:From>"7125550123" <7125550123></Email:From>
  <Email:Subject>Voice Mail from 7125550123 (3 seconds)</Email:Subject>
  <Email:DateReceived>2007-11-06T23:42:16.829Z</Email:DateReceived>
  <Email:DisplayTo>Device User</Email:DisplayTo>
  <Email:ThreadTopic>Voice Mail from 7125550123 (3 seconds)</Email:ThreadTopic>
  <Email:Importance>1</Email:Importance>
  <Email:Read>1</Email:Read>
  <AirSyncBase:Attachments>
    <AirSyncBase:Attachment>
      <AirSyncBase:DisplayName>7125550123 (3 seconds) Voice
Mail.wma</AirSyncBase:DisplayName>
      <AirSyncBase:FileReference>20%3a2%3a0</AirSyncBase:FileReference>
      <AirSyncBase:Method>1</AirSyncBase:Method>
      <AirSyncBase:EstimatedDataSize>9025</AirSyncBase:EstimatedDataSize>
      <Email2:UmAttOrder>1</Email2:UmAttOrder>
      <Email2:UmAttDuration>3</Email2:UmAttDuration>
    </AirSyncBase:Attachment>
  </AirSyncBase:Attachments>
  <AirSyncBase:Body>
    <AirSyncBase:Type>3</AirSyncBase:Type>
    <AirSyncBase:EstimatedDataSize>1512</AirSyncBase:EstimatedDataSize>
    <AirSyncBase:Truncated>1</AirSyncBase:Truncated>
  </AirSyncBase:Body>
  <Email:MessageClass>IPM.Note.Microsoft.Voicemail.UM.CA</Email:MessageClass>
  <Email:InternetCPID>20127</Email:InternetCPID>
  <Email:ContentClass>urn:content-classes:message</Email:ContentClass>
  <AirSyncBase:NativeBodyType>3</AirSyncBase:NativeBodyType>
  <Email2:CallerID>7125550123</Email2:CallerID>
  <Email2:UserNotes>7125550123</Email2:UserNotes>
</ApplicationData>
</Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.6 Example Sync Response Adding a Text Attachment

The following example shows the **Sync** command response adding one e-mail item with a text attachment to the client.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 04 Mar 2009 22:48:48 GMT
Content-Length: 444

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:AirSyncBase="AirSyncBase:" xmlns:Email2="POOMMAIL2:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>334239291</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
  <Commands>

```

```

<Add>
  <ServerId>5:3</ServerId>
  <ApplicationData>
    <Email:To>>"Device User" &lt;deviceuser@example.com>></Email:To>
    <Email:From>"Device User2" &lt;deviceuser2@example.com>></Email:From>
    <Email:Subject>With Attachment</Email:Subject>
    <Email:DateReceived>2009-03-04T22:48:41.211Z</Email:DateReceived>
    <Email:DisplayTo>Device User</Email:DisplayTo>
    <Email:ThreadTopic>With Attachment</Email:ThreadTopic>
    <Email:Importance>1</Email:Importance>
    <Email:Read>0</Email:Read>
    <AirSyncBase:Attachments>
      <AirSyncBase:Attachment>
        <AirSyncBase:DisplayName>Test.txt</AirSyncBase:DisplayName>
        <AirSyncBase:FileReference>5%3a3%3a0</AirSyncBase:FileReference>
        <AirSyncBase:Method>1</AirSyncBase:Method>
        <AirSyncBase:EstimatedDataSize>84</AirSyncBase:EstimatedDataSize>
      </AirSyncBase:Attachment>
    </AirSyncBase:Attachments>
    <AirSyncBase:Body>
      <AirSyncBase:Type>1</AirSyncBase:Type>
      <AirSyncBase:EstimatedDataSize>33</AirSyncBase:EstimatedDataSize>
      <AirSyncBase:Truncated>1</AirSyncBase:Truncated>
    </AirSyncBase:Body>
    <Email:MessageClass>IPM.Note</Email:MessageClass>
    <Email:InternetCPID>20127</Email:InternetCPID>
    <Email:Flag />
    <Email:ContentClass>urn:content-classes:message</Email:ContentClass>
    <AirSyncBase:NativeBodyType>1</AirSyncBase:NativeBodyType>
    <Email2:ConversationId>YgE□:ConversationId>YgE□ent-clEmail2:ConversationId>
    <Email2:ConversationIndex>É□:ConversatEmail2:ConversationIndex>
    <Email:Categories />
  </ApplicationData>
</Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.7 Example Sync Request Deleting One E-mail

The following example shows the **Sync** command request sent by the client to remove the e-mail message described in section [4.1.4](#) from the server.

```

POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1174511196</SyncKey>
      <CollectionId>5</CollectionId>
    </Collection>
  </Collections>
</Sync>

```

```

    <DeletesAsMoves>1</DeletesAsMoves>
    <GetChanges>1</GetChanges>
    <WindowSize>512</WindowSize>
    <Commands>
      <Delete>
        <ServerId>5:10</ServerId>
      </Delete>
    </Commands>
  </Collection>
</Collections>
</Sync>

```

4.1.8 Example Success Sync Response

The following example shows the **Sync** command response sent by the server after completing the deletion requested in section [4.1.7](#).

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 05 Feb 2009 00:10:43 GMT
Content-Length: 33

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>721953595</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>

```

4.1.9 Example Sync Response Adding a Meeting Request

The following example shows the **Sync** command response sent by the server to add a meeting request to the Inbox.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 19 Feb 2009 08:35:28 GMT
Content-Length: 1538

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:AirSyncBase="AirSyncBase:" xmlns:Email2="POOMMAIL2:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1419832287</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>5:13</ServerId>
          <ApplicationData>

```

```

<Email:To>"Device User" &lt;deviceuser@example.com>;</Email:To>
<Email:From>"Device User2" &lt;deviceuser2@example.com>;</Email:From>
<Email:Subject>Example Meeting Request</Email:Subject>
<Email:DateReceived>2009-02-19T08:35:17.922Z</Email:DateReceived>
<Email:DisplayTo>Device User</Email:DisplayTo>
<Email:ThreadTopic>Example Meeting Request</Email:ThreadTopic>
<Email:Importance>1</Email:Importance>
<Email:Read>0</Email:Read>
<AirSyncBase:Body>
  <AirSyncBase:Type>3</AirSyncBase:Type>
  <AirSyncBase:EstimatedDataSize>437</AirSyncBase:EstimatedDataSize>
  <AirSyncBase:Truncated>1</AirSyncBase:Truncated>
</AirSyncBase:Body>
<Email:MessageClass>IPM.Schedule.Meeting.Request</Email:MessageClass>
<Email:MeetingRequest>
  <Email:AllDayEvent>0</Email:AllDayEvent>
  <Email:StartTime>2009-02-20T15:30:00.000Z</Email:StartTime>
  <Email:DtStamp>2009-02-19T08:35:15.786Z</Email:DtStamp>
  <Email:EndTime>2009-02-20T16:30:00.000Z</Email:EndTime>
  <Email:InstanceType>0</Email:InstanceType>
  <Email:Location>Cafe</Email:Location>
  <Email:Organizer>"Device User2"
    &lt;deviceuser2@example.com>;</Email:Organizer>
  <Email:Reminder>900</Email:Reminder>
  <Email:ResponseRequested>1</Email:ResponseRequested>
  <Email:Sensitivity>0</Email:Sensitivity>
  <Email:BusyStatus>2</Email:BusyStatus>

  <Email:TimeZone>aAEAAcGARwBNAFQALQAwADYAOGAwADAAKQAgAEMAZQBwAHQAcgBhAGwAIABUAGkAbQB1CAAKABVA
  FMAIAAmCAAQwAAAsAAAABAAIAAAAAAAAAAAAAAAcGARwBNAFQALQAwADYAOGAwADAAKQAgAEMAZQBwAHQAcgBhAGwAIA
  BUAGkAbQB1CAAKABVAFMAIAAmCAAQwAAAMAAACAAIAAAAAAAAAAXP//w==</Email:TimeZone>

  <Email:GlobalObjId>BAAAAIAA4AB0xbCQGOLgCAAAAADYSxf9bJLJAQAAAAAAAAAAEAAAAJEHL7SUox5GtgQV1TYDY4
  A=</Email:GlobalObjId>
  </Email:MeetingRequest>
  <Email:InternetCPID>28591</Email:InternetCPID>
  <Email:Flag />
  <Email:ContentClass>urn:content-classes:calendarmessage</Email:ContentClass>
  <AirSyncBase:NativeBodyType>3</AirSyncBase:NativeBodyType>
  <Email2:ConversationId>Lð0+*û@â²&#x1A; &#x15;EñM0±</Email2:ConversationId>
  <Email2:ConversationIndex>É'lý&#x1F;</Email2:ConversationIndex>
  <Email:Categories />
</ApplicationData>
</Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.10 Example Sync Response Adding a Meeting Request with a Recurrence

The following example shows the **Sync** command response sent by the server to add a meeting request to the Inbox. This meeting occurs every month (**Type** is 3 and **Interval** is 1), in the third week of the month (**WeekOfMonth** is 3), on Tuesday (**DayOfWeek** is 4).

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 19 Feb 2009 08:47:27 GMT

```

Content-Length: 950

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:AirSyncBase="AirSyncBase:" xmlns:Email2="POOMMAIL2:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>2086787787</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>5:14</ServerId>
          <ApplicationData>
            <Email:To>"Device User" &lt;deviceuser@example.com&gt;</Email:To>
            <Email:From>"Device User2" &lt;deviceuser2@example.com&gt;</Email:From>
            <Email:Subject>Monthly Meeting</Email:Subject>
            <Email:DateReceived>2009-02-19T08:47:21.842Z</Email:DateReceived>
            <Email:DisplayTo>Device User</Email:DisplayTo>
            <Email:ThreadTopic>Monthly Meeting</Email:ThreadTopic>
            <Email:Importance>1</Email:Importance>
            <Email:Read>0</Email:Read>
            <AirSyncBase:Body>
              <AirSyncBase:Type>3</AirSyncBase:Type>
              <AirSyncBase:EstimatedDataSize>503</AirSyncBase:EstimatedDataSize>
              <AirSyncBase:Truncated>1</AirSyncBase:Truncated>
            </AirSyncBase:Body>
            <Email:MessageClass>IPM.Schedule.Meeting.Request</Email:MessageClass>
            <Email:MeetingRequest>
              <Email:AllDayEvent>0</Email:AllDayEvent>
              <Email:StartTime>2009-03-17T20:00:00.000Z</Email:StartTime>
              <Email:DtStamp>2009-02-19T08:47:19.527Z</Email:DtStamp>
              <Email:EndTime>2009-03-17T21:00:00.000Z</Email:EndTime>
              <Email:InstanceType>1</Email:InstanceType>
              <Email:Location>My Office</Email:Location>
              <Email:Organizer>"Device User2"
&lt;deviceuser2@example.com&gt;</Email:Organizer>
              <Email:Reminder>900</Email:Reminder>
              <Email:ResponseRequested>1</Email:ResponseRequested>
              <Email:Recurrences>
                <Email:Recurrence>
                  <Email:Type>3</Email:Type>
                  <Email:Interval>1</Email:Interval>
                  <Email:Until>20091229T210000Z</Email:Until>
                  <Email:WeekOfMonth>3</Email:WeekOfMonth>
                  <Email:DayOfWeek>4</Email:DayOfWeek>
                </Email:Recurrence>
              </Email:Recurrences>
              <Email:Sensitivity>0</Email:Sensitivity>
              <Email:BusyStatus>2</Email:BusyStatus>

            <Email:TimeZone>aAEAAcGARwBNAFQALQAwADYAOGAwADAAKQAgAEMAZQBwAHQAcgBhAGwAIABUAGkAbQBlCAAKABVA
FMAIAAAmCAAQwAAAsAAAAABAAIAAAAAAAAAAAAAAAcGARwBNAFQALQAwADYAOGAwADAAKQAgAEMAZQBwAHQAcgBhAGwAIA
BUAGkAbQBlCAAKABVAFMAIAAAmCAAQwAAAMAAAAACAIAAAAAAAAAAAxP//w==</Email:TimeZone>

            <Email:GlobalObjId>BAAAAIA4AB0xbCQGoLgCAAAAAADok5WnbpLJAQAAAAAAAAAAAAEAAAAAP4Ao5IYwQdKiFkDBeGTTg
Y=</Email:GlobalObjId>
          </Email:MeetingRequest>
          <Email:InternetCPID>28591</Email:InternetCPID>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```



```

        <Email:Flag />
        <Email:ContentClass>urn:content-classes:calendarmessage</Email:ContentClass>
        <AirSyncBase:NativeBodyType>3</AirSyncBase:NativeBodyType>
    <Email2:ConversationId>'MÅ' &amp;Kä °V÷ŽÓ&#x16;xû</Email2:ConversationId>
        <Email2:ConversationIndex>Ě' n¬,</Email2:ConversationIndex>
        <Email:Categories />
    </ApplicationData>
</Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.11 Example ItemOperations Request

The following example shows the **ItemOperations** command request to retrieve the metadata and body of the e-mail message whose ServerId value is 5:10.

```

POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:AirSync="AirSync:" xmlns:AirSyncBase="AirSyncBase:"
xmlns="ItemOperations:">
    <Fetch>
        <Store>Mailbox</Store>
        <AirSync:CollectionId>5</AirSync:CollectionId>
        <AirSync:ServerId>5:10</AirSync:ServerId>
        <Options>
            <AirSyncBase:BodyPreference>
                <AirSyncBase:Type >2</AirSyncBase:Type>
            </AirSyncBase:BodyPreference>
        </Options>
    </Fetch>
</ItemOperations>

```

4.1.12 Example ItemOperations Response With Fetched Content

The following example shows the **ItemOperations** command response to the request in section [4.1.11](#).

Note that in the example, the data in the <data> string is XML escaped (NOT HTML). However, as these values are passed over the wire using WBXML, they are passed unencoded (ie < and >).

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 19 Feb 2009 01:44:09 GMT
Content-Length: 753

<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:AirSync="AirSync:" xmlns:Email="POOMMAIL:"
xmlns:AirSyncBase="AirSyncBase:" xmlns:Email2="POOMMAIL2:" xmlns="ItemOperations:">

```

```

<Status>1</Status>
<Response>
  <Fetch>
    <Status>1</Status>
    <AirSync:CollectionId>5</AirSync:CollectionId>
    <AirSync:ServerId>5:10</AirSync:ServerId>
    <AirSync:Class>Email</AirSync:Class>
    <Properties>
      <Email:To>"Device User" &lt;deviceuser@example.com&gt;</Email:To>
      <Email:From>"Device User2" &lt;deviceuser2@example.com&gt;</Email:From>
      <Email:Subject>Fetch this content.</Email:Subject>
      <Email:DateReceived>2009-02-19T01:43:25.266Z</Email:DateReceived>
      <Email:DisplayTo>Device User</Email:DisplayTo>
      <Email:ThreadTopic>Fetch this content.</Email:ThreadTopic>
      <Email:Importance>1</Email:Importance>
      <Email:Read>0</Email:Read>
      <AirSyncBase:Body>
        <AirSyncBase:Type>2</AirSyncBase:Type>
        <AirSyncBase:EstimatedDataSize>376</AirSyncBase:EstimatedDataSize>
        <AirSyncBase:Data>&lt;html dir="ltr"&gt;
&lt;head&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8"&gt;
&lt;style&gt;&lt;/style&gt;&lt;style id="owaParaStyle"&gt;
&lt;!--
p
{margin-top:0px;
margin-bottom:0px}
--&gt;
&lt;/style&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;div style="font-size:13px; color:#000000; direction:ltr; font-family:Tahoma"&gt;
&lt;div&gt;This is the content that was truncated.&lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
&lt;/AirSyncBase:Data>
      </AirSyncBase:Body>
      <Email:MessageClass>IPM.Note</Email:MessageClass>
      <Email:InternetCPID>28591</Email:InternetCPID>
      <Email:Flag />
      <Email:ContentClass>urn:content-classes:message</Email:ContentClass>
      <AirSyncBase:NativeBodyType>2</AirSyncBase:NativeBodyType>
      <Email2:ConversationId>€%ÿ&#x18; &#x8; ;B~</Email2:ConversationId>
      <Email2:ConversationIndex>&#x18; &#x8;</Email2:ConversationIndex>
    </Properties>
  </Fetch>
</Response>
</ItemOperations>

```

4.2 Setting MeetingRequest Recurrence Intervals

The following examples show how to use the recurrence properties to set meetings on different days of the month.

To set a recurrence to occur on the 15th day of every month, use the following values:

MeetingRequest.Recurrences.Recurrence.Type = 2

MeetingRequest.Recurrences.Recurrence.Interval = 1

MeetingRequest.Recurrences.Recurrence.DayOfMonth = 15

To set a recurrence to occur on the 31st day of every other month, use the following values:

MeetingRequest.Recurrences.Recurrence.Type = 2

MeetingRequest.Recurrences.Recurrence.Interval = 2

MeetingRequest.Recurrences.Recurrence.DayOfMonth = 31

To set a recurrence to occur every weekday for one week, use the following values:

MeetingRequest.Recurrences.Recurrence.Type = 0

MeetingRequest.Recurrences.Recurrence.Interval = 1

MeetingRequest.Recurrences.Recurrence.Occurrences = 5

MeetingRequest.Recurrences.Recurrence.DayOfWeek = 62

4.3 Setting Flags on the Client and Server

This section provides an example request and response message that are related to setting flags on the client and server.

Note the following:

- Implicit deletes: This term means that if a tag is not found in the **Flag** container, then the property is deleted.
- Although there are tokens from the **Tasks** namespace, all elements are saved on the e-mail item only. No task items are created.

4.3.1 Example Sync Request Setting a Flag on the Client

The following example **Sync** command request sets a flag with a start date, due date, but no reminder.

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:Tasks="POOMTASKS:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1520171944</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Commands>
        <Change>
```

```

    <ServerId>5:3</ServerId>
    <ApplicationData>
      <Email:Read>1</Email:Read>
      <Email:Flag>
        <Email:Status>2</Email:Status>
        <Email:FlagType>for Follow Up</Email:FlagType>
        <Tasks:StartDate>2009-02-24T08:00:00.000Z</Tasks:StartDate>
        <Tasks:UtcStartDate>2009-02-24T08:00:00.000Z</Tasks:UtcStartDate>
        <Tasks:DueDate>2009-02-25T12:00:00.000Z</Tasks:DueDate>
        <Tasks:UtcDueDate>2009-02-25T12:00:00.000Z</Tasks:UtcDueDate>
        <Tasks:ReminderSet>0</Tasks:ReminderSet>
      </Email:Flag>
    </ApplicationData>
  </Change>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.3.2 Example Sync Request Setting a Flag on the Server

The following example **Sync** command response sets a flag with a start date, due date, and a reminder.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Thu, 19 Feb 2009 01:20:24 GMT
Content-Length: 245

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:Tasks="POOMTASKS:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>735431712</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>5:7</ServerId>
          <ApplicationData>
            <Email:Flag>
              <Tasks:DueDate>2009-02-20T08:00:00.000Z</Tasks:DueDate>
              <Tasks:UtcDueDate>2009-02-20T08:00:00.000Z</Tasks:UtcDueDate>
              <Tasks:UtcStartDate>2009-02-19T08:00:00.000Z</Tasks:UtcStartDate>
              <Tasks:Subject>With Reminder</Tasks:Subject>
              <Email:Status>2</Email:Status>
              <Email:FlagType>Flag for follow up</Email:FlagType>
              <Tasks:StartDate>2009-02-19T08:00:00.000Z</Tasks:StartDate>
              <Tasks:ReminderSet>1</Tasks:ReminderSet>
              <Tasks:ReminderTime>2009-02-19T21:00:00.000Z</Tasks:ReminderTime>
            </Email:Flag>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>

```

</Sync>

4.3.3 Example Sync Request Setting the Complete Flag

The following is an example of how to send the request message to the server when the Completed flag for a **FlagType** set to **Flag for follow up** has been selected on the client.

Note that the **DateCompleted** element indicates when the user selected the **Completed** flag in Outlook. The **CompleteTime** element indicates the time that the item was marked as finished.

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:Tasks="POOMTASKS:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>509846121</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Commands>
        <Change>
          <ServerId>5:5</ServerId>
          <ApplicationData>
            <Email:Read>1</Email:Read>
            <Email:Flag>
              <Email:Status>1</Email:Status>
              <Email:FlagType>Flag for follow up</Email:FlagType>
              <Email:CompleteTime>2009-02-19T08:30:00.000Z</Email:CompleteTime>
              <Tasks:StartDate>2009-02-19T08:00:00.000Z</Tasks:StartDate>
              <Tasks:UtcStartDate>2009-02-19T08:00:00.000Z</Tasks:UtcStartDate>
              <Tasks:DueDate>2009-02-19T08:00:00.000Z</Tasks:DueDate>
              <Tasks:UtcDueDate>2009-02-19T08:00:00.000Z</Tasks:UtcDueDate>
              <Tasks:DateCompleted>2009-02-19T09:30:00.000Z</Tasks:DateCompleted>
              <Tasks:ReminderSet>0</Tasks:ReminderSet>
              <Tasks:ReminderTime>2009-02-24T20:00:00.000Z</Tasks:ReminderTime>
              <Tasks:Subject>Please follow up</Tasks:Subject>
            </Email:Flag>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.3.4 Example Sync Request Clearing a Flag on the Client

The following example shows what to include in a request message to clear a flag.

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v14Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: B-290.test.contoso.com
```

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" xmlns:Tasks="POOMTASKS:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1401532757</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Commands>
        <Change>
          <ServerId>5:5</ServerId>
          <ApplicationData>
            <Email:Read>1</Email:Read>
            <Email:Flag>
              <Email:Status>0</Email:Status>
              <Email:FlagType>Flag for follow up</Email:FlagType>
              <Email:CompleteTime>2009-02-19T08:30:00.000Z</Email:CompleteTime>
              <Tasks:StartDate>2009-02-19T08:00:00.000Z</Tasks:StartDate>
              <Tasks:UtcStartDate>2009-02-19T08:00:00.000Z</Tasks:UtcStartDate>
              <Tasks:DueDate>2009-02-19T08:00:00.000Z</Tasks:DueDate>
              <Tasks:UtcDueDate>2009-02-19T08:00:00.000Z</Tasks:UtcDueDate>
              <Tasks:DateCompleted>2009-02-20T09:30:00.000Z</Tasks:DateCompleted>
              <Tasks:ReminderSet>0</Tasks:ReminderSet>
              <Tasks:ReminderTime>2009-02-24T20:00:00.000Z</Tasks:ReminderTime>
              <Tasks:Subject>Please follow up</Tasks:Subject>
            </Email:Flag>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010

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

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

[<1> Section 2.1:](#) Exchange 2007 does not support the Email2 namespace. The Email2 namespace is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<2> Section 2.2.1.8:](#) The Categories type is not supported by Exchange 2007. The Categories type is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<3> Section 2.2.2.12:](#) Exchange 2007 does not support the UmAttOrder element. The UmAttOrder element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<4> Section 2.2.2.13:](#) Exchange 2007 does not support the UmAttDuration element. The UmAttDuration element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<5> Section 2.2.2.20:](#) Outlook 2007 and Outlook 2010 limit the value of the MeetingRequest.Location element to 255.

[<6> Section 2.2.2.33:](#) Exchange 2007 does not support the CalendarType element. The CalendarType element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<7> Section 2.2.2.34:](#) Exchange 2007 does not support the IsLeapMonth element. The IsLeapMonth element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<8> Section 2.2.2.39:](#) Exchange 2007 does not support the DisallowNewTimeProposal element. The DisallowNewTimeProposal element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<9> Section 2.2.2.56:](#) Exchange 2007 does not support the UmCallerId element. The UmCallerID element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<10> Section 2.2.2.57:](#) Exchange 2007 does not support the UmUserNotes element. The UmUserNotes element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<11> Section 2.2.2.58:](#) Exchange 2007 does not support the <ConversationId> element. The <ConversationId> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<12> Section 2.2.2.59:](#) Exchange 2007 does not support the ConversationIndex element. The ConversationIndex element is only supported when the MS-ASProtocolVersion header is set to 14.0.

[<13> Section 2.2.2.60:](#) Exchange 2007 does not support the LastVerbExecuted element. The LastVerbExecuted element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<14> Section 2.2.2.61:](#) Exchange 2007 does not support the LastVerbExecutionTime element. The LastVerbExecutionTime element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<15> Section 2.2.2.62:](#) Exchange 2007 does not support the ReceivedAsBcc element. The ReceivedAsBcc element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<16> Section 2.2.2.63:](#) Exchange 2007 does not support the Sender element. The Sender element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

[<17> Section 2.2.2.64:](#) The Category element is not supported by Exchange 2007. The Category element is only supported when the MS-ASProtocolVersion header value is set to 14.0.

7 Change Tracking

This section identifies changes made to [MS-ASEMAIL] protocol documentation between November 2009 and February 2010 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

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

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

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

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

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

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2 Message Syntax	50742 Added string length restrictions to <To>, <From>, and <Location>.	N	Content update.
2.2.1 Complex Types	Added the Categories type to the table.	N	Content update.
2.2.1 Complex Types	52513 Added links to all the complex type topics.	N	Content update.
2.2.2 Elements	50716 Revised the description of the Occurrences element to clarify that it supports a series of recurring meetings.	N	Editorially updated.
2.2.2 Elements	50737 Added that the Sync <Status> value of 6 is returned when child e-mail elements are incorrectly included in a Sync request.	Y	Content update.
2.2.2 Elements	Added Categories.Category to the table of elements.	N	Content update.
2.2.2 Elements	52513 Added links to all element topics.	N	Content update.
2.2.2.1	50742	Y	Content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
To	Changed maximum string length to 32,768.		update.
2.2.2.3 From	50742 Changed maximum string length value to 32,768.	Y	Content update.
2.2.2.12 Attachments.Attachment.UmAttOrder	50748 Added normative language to indicate that the value of the most recent attachment in an e-mail item must have a UmAttOrder value of 1.	N	Content update.
2.2.2.20 MeetingRequest.Location	47882 Added maximum character length.	Y	Content update.
2.2.2.22 MeetingRequest.RecurrenceId	50729 Clarified that the <RecurrenceId> element is only included in server response messages.	Y	Content update.
2.2.2.28 MeetingRequest.Recurrences.Recurrence.Occurrences	50716 Clarified that the recurring meetings are a series.	N	Editorially updated.
2.2.2.37 MeetingRequest.TimeZone	50714 Clarified that this element has a String datatype, with a TimeZone format.	N	Content update.
2.2.2.46 Flag.StartDate	50783 Added <Status> value returned by Sync response if <Flag.StartDate> occurs after <Flag.DueDate>.	Y	Content update.
2.2.2.46 Flag.StartDate	50784 Added <Status> value returned by Sync response if <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> are not all set, or not all NULL.	Y	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.2.47 Flag.DueDate	50783 Added <Status> value returned by Sync response if <Flag.DueDate> occurs before <Flag.StartDate>.	Y	Content update.
2.2.2.47 Flag.DueDate	50784 Added <Status> value returned by Sync response if <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> are not all set, or not all NULL.	Y	Content update.
2.2.2.48 Flag.UTCStartDate	50783 Added <Status> value returned by Sync response if <Flag.UTCStartDate> occurs after <Flag.UTCStartDate>.	Y	Content update.
2.2.2.48 Flag.UTCStartDate	50784 Added <Status> value returned by Sync response if <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> are not all set, or not all NULL.	Y	Content update.
2.2.2.49 Flag.UTCDueDate	50783 Added <Status> value returned by Sync response if <Flag.UTCDueDate> occurs after <Flag.UTCStartDate>.	Y	Content update.
2.2.2.49 Flag.UTCDueDate	50784 Added <Status> value returned by Sync response if <Flag.StartDate>, <Flag.DueDate>, <Flag.UTCStartDate>, and <Flag.UTCDueDate> are not all set, or not all	Y	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
	NULL.		
2.2.2.51 Flag.ReminderTime	50797 Added negative behavior when <Flag.ReminderSet> is TRUE (1) and <Flag.ReminderTime> is not included in a Sync request.	Y	Content update.
2.2.2.56 UmCallerID	50803 Added that <Status> value 6 is returned in a Sync response if the client attempts to send this value to the server.	Y	Content update.
2.2.2.57 UmUserNotes	50803 Added that <Status> value 6 is returned in a Sync response if the client attempts to send this value to the server.	Y	Content update.
2.2.2.58 ConversationId	50807 Added a link to [MS-ASCON] for more information about conversation.	N	Content update.
2.2.2.58 ConversationId	50817 Added that <Status> value 6 is returned in a Sync response if the client attempts to change the <ConversationId> value.	Y	Content update.
2.2.2.59 ConversationIndex	50817 Added that <Status> value 6 is returned in a Sync response if the client attempts to change the <ConversationIndex> value.	Y	Content update.
3.1.5.2 Search Command Request	50820 Added the <Status> value is returned if E-mail class complex types and elements are included in a Search request.	Y	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
3.2.4.5 Send Changes to the Client	50824 Identified what the server MUST and SHOULD send to the client depending on the changes made to the E-mail item.	N	Content update.
3.2.4.5 Send Changes to the Client	Changed title to reflect that this section discusses all changes sent to the client.	N	Content update.

8 Index

A

[Applicability](#) 8

C

[Capability negotiation](#) 8

[Change tracking](#) 58

Client

[overview](#) 33

E

[Examples - overview](#) 40

F

[Fields – vendor-extensible](#) 8

G

[Glossary](#) 6

I

[Implementer – security considerations](#) 55

[Index of security parameters](#) 55

[Informative references](#) 7

[Introduction](#) 6

M

Message

[transport](#) 9

Messages

[overview](#) 9

N

[Normative references](#) 6

O

[Overview](#) 7

P

[Parameters – security index](#) 55

[Preconditions](#) 7

[Prerequisites](#) 7

[Product behavior](#) 56

R

References

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

S

Security

[implementer considerations](#) 55

[overview](#) 55

[parameter index](#) 55

Server

[overview](#) 36

[Standards Assignments](#) 8

T

[Tracking changes](#) 58

[Transport](#) 9

V

[Vendor-extensible fields](#) 8

[Versioning](#) 8