

[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.mspx>). 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.

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		Initial Release.
02/04/2009	1.01		Revised and edited technical content.
03/04/2009	1.02		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for 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	7
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	11
2.2.1.1 Attachments	12
2.2.1.2 Attachments.Attachment	12
2.2.1.3 Body	12
2.2.1.4 MeetingRequest	12
2.2.1.5 MeetingRequest.Recurrences	12
2.2.1.6 MeetingRequest.Recurrences.Recurrence	12
2.2.1.7 MeetingRequest.Categories	13
2.2.1.8 Flag	13
2.2.2 Elements	13
2.2.2.1 To	16
2.2.2.2 Cc	16
2.2.2.3 From	16
2.2.2.4 Subject	17
2.2.2.5 ReplyTo	17
2.2.2.6 DateReceived	17
2.2.2.7 DisplayTo	17
2.2.2.8 ThreadTopic	17
2.2.2.9 Importance	17
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	18
2.2.2.14 MessageClass	18
2.2.2.15 MeetingRequest.AllDayEvent	20
2.2.2.16 MeetingRequest.StartTime	20
2.2.2.17 MeetingRequest.DtStamp	20
2.2.2.18 MeetingRequest.EndTime	20
2.2.2.19 MeetingRequest.InstanceType	20
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	21
2.2.2.24 MeetingRequest.ResponseRequested	21

2.2.2.25	MeetingRequest.Recurrences.Recurrence.Type	21
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	22
2.2.2.29	MeetingRequest.Recurrences.Recurrence.WeekOfMonth	22
2.2.2.30	MeetingRequest.Recurrences.Recurrence.DayOfMonth	22
2.2.2.31	MeetingRequest.Recurrences.Recurrence.DayOfWeek	23
2.2.2.32	MeetingRequest.Recurrences.Recurrence.MonthOfYear	23
2.2.2.33	MeetingRequest.Sensitivity	23
2.2.2.34	MeetingRequest.BusyStatus	24
2.2.2.35	MeetingRequest.TimeZone	24
2.2.2.36	MeetingRequest.GlobalObjId	24
2.2.2.37	MeetingRequest.Categories.Category	24
2.2.2.38	MeetingRequest.Categories.DisallowNewTimeProposal	25
2.2.2.39	InternetCPID	25
2.2.2.40	Flag.Subject	25
2.2.2.41	Flag.Status	25
2.2.2.42	Flag.FlagType	25
2.2.2.43	Flag.DateCompleted	26
2.2.2.44	Flag.CompleteTime	26
2.2.2.45	Flag.StartDate	26
2.2.2.46	Flag.DueDate	26
2.2.2.47	Flag.UTCStartDate	27
2.2.2.48	Flag.UTCDueDate	27
2.2.2.49	Flag.ReminderSet	27
2.2.2.50	Flag.ReminderTime	28
2.2.2.51	Flag.OrdinalDate	28
2.2.2.52	Flag.SubOrdinalDate	28
2.2.2.53	ContentClass	28
2.2.2.54	NativeBodyType	28
2.2.2.55	UmCallerID	28
2.2.2.56	UmUserNotes	29
2.2.2.57	ConversationId	29
2.2.2.58	ConversationIndex	30
2.2.2.59	LastVerbExecuted	30
2.2.2.60	LastVerbExecutionTime	31
2.2.2.61	ReceivedAsBcc	31
2.2.2.62	Sender	31

3	Protocol Details	32
3.1	Client Details	32
3.1.1	Abstract Data Model	32
3.1.2	Timers	32
3.1.3	Initialization	32
3.1.4	Higher-Layer Triggered Events	32
3.1.4.1	Synchronize E-mail Between Client and Server	32
3.1.4.2	Search E-mail	32
3.1.4.3	Retrieve Individual E-mail	32
3.1.4.4	Send Flag Updates to the Server	32
3.1.5	Message Processing Events and Sequencing Rules	34
3.1.5.1	ItemOperations Command Request	34
3.1.5.2	Search Command Request	34
3.1.5.3	Sync Command Request	34

3.1.6	Timer Events.....	35
3.1.7	Other Local Events	35
3.2	Server Details	35
3.2.1	Abstract Data Model.....	35
3.2.2	Timers	35
3.2.3	Initialization	35
3.2.4	Higher-Layer Triggered Events	35
3.2.4.1	Synchronize E-mail Between Client and Server.....	35
3.2.4.2	Search E-mail	35
3.2.4.3	Retrieve Individual E-mail	36
3.2.4.4	Change Tracking Semantics for Flagging Properties	36
3.2.4.5	Send Flagged Changes to the Client.....	36
3.2.5	Message Processing Events and Sequencing Rules	37
3.2.5.1	ItemOperations Command Response	37
3.2.5.2	Search Command Response	37
3.2.5.3	Sync Command Response	37
3.2.6	Timer Events.....	38
3.2.7	Other Local Events	38
4	Protocol Examples	39
4.1	Synchronizing HTML E-Mail.....	39
4.1.1	Example Sync Request for Inbox with HTML Mail Support	39
4.1.2	Example Sync Request for Inbox with Body Preferences.....	39
4.1.3	Example Sync Response for E-Mail with One HTML Message.....	40
4.1.4	Example ItemOperations Request.....	41
4.1.5	Example ItemOperations Response With Fetched Content.....	41
4.1.6	Example Sync Response Adding an Electronic Voice Mail Attachment	43
4.1.7	Example Sync Response Adding a Text Attachment	44
4.1.8	Example Sync Request Deleting One E-mail.....	45
4.1.9	Example Sync Response Adding a Meeting Request	45
4.1.10	Example Sync Response Adding a Meeting Request with a Recurrence	46
4.1.11	Example Success Sync Response	48
4.2	Setting MeetingRequest Recurrence Intervals	48
4.3	Setting Flags on the Client and Server.....	49
4.3.1	Example Sync Request Setting a Flag on the Client.....	49
4.3.2	Example Sync Request Setting a Flag on the Server	50
4.3.3	Example Sync Request Setting the Complete Flag.....	50
4.3.4	Example Sync Request Clearing a Flag on the Client.....	51
5	Security.....	53
5.1	Security Considerations for Implementers.....	53
5.2	Index of Security Parameters	53
6	Appendix A: Product Behavior	54
7	Change Tracking	55
8	Index.....	56

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 [\[MS-OXGLOS\]](#):

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 Protocol Specification](#)", December 2008.

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

[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 2822, October 2008, <http://www.ietf.org/rfc/rfc5322.txt>.

[XML] Bray, T., et al., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", <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 **Sync** command.

```
<?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:">
  <xs:import namespace="AirSyncBase" />
  <xs:element name="To" type="xs:string" />
  <xs:element name="Cc" type="xs:string" />
  <xs:element name="From" type="xs:string" />
  <xs:element name="Subject" type="xs:string" />
  <xs:element name="ReplyTo" type="xs:string" />

  <xs:element name="DateReceived" type="xs:dateTime" />
  <xs:element name="DisplayTo" type="xs:string" />
  <xs:element name="ThreadTopic" type="xs:string" />
  <xs:element name="Importance" type="xs:unsignedByte" />
  <xs:element name="Read" type="xs:unsignedByte" />
  <xs:element name="Attachments" type="A:Attachments" >
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Attachment">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DisplayName" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Body" type="A:Body" />
  <xs:element name="MessageClass" type="xs:string" />
  <xs:element name="MeetingRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AllDayEvent" type="xs:unsignedByte" />
        <xs:element name="StartTime" type="xs:dateTime" />
        <xs:element name="DtStamp" type="xs:dateTime" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:element name="EndTime" type="xs:dateTime" />
<xs:element name="InstanceType" type="xs:unsignedByte" />
<xs:element minOccurs="0" name="Location" type="xs:string" />
<xs:element name="Organizer" type="xs:string" />
<xs:element minOccurs="0" name="RecurrenceId" type="xs:dateTime" />
<xs:element name="Reminder" type="xs:unsignedShort" />
<xs:element name="ResponseRequested" type="xs:unsignedByte" />
<xs:element name="Recurrences">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Recurrence">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Type"
type="xs:unsignedByte" />
            <xs:element name="Interval"
type="xs:integer" />
            <xs:element name="Until" type="xs:datetime"
/>
            <xs:element name="Occurrences"
type="xs:integer" />
            <xs:element name="WeekOfMonth"
type="xs:integer" />
            <xs:element name="DayOfMonth"
type="xs:integer" />
            <xs:element name="DayOfWeek"
type="xs:integer" />
            <xs:element name="MonthOfYear"
type="xs:integer" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Sensitivity" type="xs:integer" />
<xs:element name="BusyStatus" type="xs:integer" />
<xs:element name="TimeZone" type="xs:string" />
<xs:element name="GlobalObjId" type="xs:string" />
<xs:element name="Categories">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="300" name="Category"
type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="DisallowNewTimeProposal" type="xs:unsignedByte" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="InternetCPID" type="xs:string" />
<xs:element name="Flag">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Subject" type="xs:string" />
      <xs:element name="Status" type="xs:integer" />
      <xs:element name="FlagType" type="xs:string" />
      <xs:element name="DateCompleted" type="xs:dateTime" />
      <xs:element name="CompleteTime" type="xs:dateTime" />
      <xs:element name="StartDate" type="xs:dateTime" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <xs:element name="DueDate" type="xs:dateTime" />
        <xs:element name="UTCStartDate" type="xs:dateTime" />
        <xs:element name="UTCEndDate" type="xs:dateTime" />
        <xs:element name="ReminderSet" type="xs:unsignedByte" />
        <xs:element name="ReminderTime" type="xs:dateTime" />
        <xs:element name="OrdinalDate" type="xs:dateTime" />
        <xs:element name="SubOrdinalDate" type="xs:dateTime" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ContentClass" type="xs:string" />
<xs:element name="NativeBodyType" type="A:NativeBodyType" /></xs:schema>

```

The XML schema definition for the Email2 class in ActiveSync is as follows.

```

<?xml version="1.0" ?>
<xs:schema xmlns:tns="Email2:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="CONTACTS2:"
xmlns:xs=http://www.w3.org/2001/XMLSchema xmlns:A="AirSyncBase:">
  <xs:import namespace="AirSyncBase" />
  <xs:element name="UmCallerID" type="xs:string" />
  <xs:element name="UmUserNotes" type="xs:string" />
  <xs:element name="UmAttOrder" type="xs:integer" />
  <xs:element name="UmAttDuration" type="xs:integer" />
  <xs:element name="ConversationId" type="xs:string" />
  <xs:element name="ConversationIndex" type="xs:string" />
  <xs:element name="LastVerbExecuted" type="xs:integer" />
  <xs:element name="LastVerbExecutionTime" type="xs:dateTime" />
  <xs:element name="ReceivedAsBcc" type="xs:unsignedByte" />
  <xs:element name="Sender" type="xs:string" />
</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.
MeetingRequest.Categories	A collection of Category elements belonging to a MeetingRequest .

Complex Type	Description
Flag	The flag associated with the item, along with the item's current status.

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 MeetingRequest.Categories

The **MeetingRequest.Categories** type is a **container** type that contains the user-selected category for this message.

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

2.2.1.8 Flag

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

If a **Flag** type is defined, then it MUST contain one or more instances the **Flag** elements specified in [2.2.2.40](#) through [2.2.2.52](#).

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.

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.

Element	Description
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.
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 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.

Element	Description
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.Categories.Category	A named label for the MeetingRequest element.
MeetingRequest.Categories.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.
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.

Element	Description
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 by From .

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 1024 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 1,024 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.Display Name

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 [<2>](#) 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 always has 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 [<3>](#) 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 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.InfoPathForm	An InfoPath form, as specified by [MS-IPFFX]
IPM.Schedule.Meeting	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 can 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 <4> 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.IPNRN	Read receipt for an S/MIME encrypted and opaque-signed message.
REPORT.IPM.NOTE.SMIME.IPNNRN	Non-read receipt for an S/MIME encrypted and opaque-signed message.

MessageClass	Description
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.
REPORT.IPM.NOTE.SMIME.MULTIPARTSIGNED.IPNRN	Read receipt for an S/MIME clear signed message.
REPORT.IPM.NOTE.SMIME.MULTIPARTSIGNED.IPNNRN	Non-read 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 **TRUE**, then the attached meeting request is an all day event.

The value of this element is a **boolean** value, as specified in [\[MS-ASDTYPE\]](#) section 2.3.

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. It is defined as an element in the Email namespace.

The maximum character length of this element's value is 1,023 characters.

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.

This element **MUST** be included if this is a recurring meeting or an exception to a recurring meeting.

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 specifies 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	Rekurs daily.
1	Rekurs weekly.
2	Rekurs monthly on the Nth day of the month.
3	Rekurs monthly.
5	Rekurs yearly on the Nth day of the Nth month each year.
6	Rekurs 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 recurring meeting series 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
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 year.

2.2.2.33 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.

2.2.2.34 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.

2.2.2.35 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 **timezone** value, as specified in [\[MS-ASDTYPE\]](#) section 2.7.

2.2.2.36 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.

2.2.2.37 MeetingRequest.Categories.Category

The **MeetingRequest.Categories.Category** element is an optional child element of the categories type that specifies the user-selected category for this message. It is defined as an element in the Email namespace.

If a **Categories** element is defined, then it MUST contain one or more **Category** elements. The names of the categories are not fixed, and custom category names are allowed.

A maximum of 300 **Category** elements are allowed per categories type.

2.2.2.38 MeetingRequest.Categories.DisallowNewTimeProposal

The **DisallowNewTimeProposal** element <5> is an optional element that indicates whether recipients can propose a new meeting time. If the value is not specified, the value defaults to FALSE, meaning that new time proposals are allowed. It is defined as an element in the Email namespace.

The value of this element is a **boolean** value, as specified in [\[MS-ASDTYPE\]](#) section 2.3.

2.2.2.39 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.

2.2.2.40 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 Email 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**.

2.2.2.41 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 2.2.1.19.2.19) 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**.

2.2.2.42 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**.

2.2.2.43 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 Email 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**.

2.2.2.44 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**.

2.2.2.45 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 Email namespace.

When a flag is being updated, **Flag.StartDate** MUST NOT occur after **Flag.DueDate**.

To set a flag, **Flag.StartDate**, **Flag.DueDate**, **Flag.UTCStartDate**, and **Flag.UTCDueDate** MUST all be set, or MUST all be NULL. If the four elements are NULL, the following three conditions MUST be met for the flag to be set:

- **Flag.Status** MUST be set to 2
- **DateCompleted** MUST be set
- **Flag.ReminderTime** MUST be set if **Flag.ReminderSet** is **TRUE** (1)

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**.

2.2.2.46 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 Email namespace.

When a flag is being updated, **Flag.DueDate** MUST NOT occur before **Flag.StartDate**.

To set a flag, **Flag.StartDate**, **Flag.DueDate**, **Flag.UTCStartDate**, and **Flag.UTCDueDate** MUST all be set, or MUST all be NULL. If the four elements are NULL, the following three conditions MUST be met for the flag to be set:

- **Flag.Status** MUST be set to 2
- **DateCompleted** MUST be set
- **Flag.ReminderTime** MUST be set if **Flag.ReminderSet** is **TRUE** (1)

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**.

2.2.2.47 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 Email namespace.

When a flag is being updated, **Flag.UTCStartDate** MUST occur before **Flag.UTCDueDate**.

To set a flag, **Flag.StartDate**, **Flag.DueDate**, **Flag.UTCStartDate**, and **Flag.UTCDueDate** MUST all be set, or MUST all be NULL. If the four elements are NULL, the following three conditions MUST be met for the flag to be set:

- **Flag.Status** MUST be set to 2
- **DateCompleted** MUST be set
- **Flag.ReminderTime** MUST be set if **Flag.ReminderSet** is **TRUE** (1)

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**.

2.2.2.48 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 Email namespace.

When a flag is being updated, **Flag.UTCDueDate** MUST NOT occur before **Flag.UTCStartDate**.

To set a flag, **Flag.StartDate**, **Flag.DueDate**, **Flag.UTCStartDate**, and **Flag.UTCDueDate** MUST all be set, or MUST all be NULL. If the four elements are NULL, the following three conditions MUST be met for the flag to be set:

- **Flag.Status** MUST be set to 2
- **DateCompleted** MUST be set
- **Flag.ReminderTime** MUST be set if **Flag.ReminderSet** is **TRUE** (1)

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**.

2.2.2.49 Flag.ReminderSet

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

A reminder MUST NOT be set if **Flag.FlagType** is set to a meeting request.

The value of this element is a **boolean** value, as specified in [\[MS-ASDTYPE\]](#) section 2.3.

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

2.2.2.50 **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 Email namespace.

Flag.ReminderTime MUST be set if **Flag.ReminderSet** is **TRUE** (1).

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**.

2.2.2.51 **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 Email 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**.

2.2.2.52 **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 Email 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**.

2.2.2.53 **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.

2.2.2.54 **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 **UmCallerID**

The **UmCallerID** element [<6>](#) 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. 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 IP (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.VoiceMail
- IPM.Note.Microsoft.VoiceMail.UM
- IPM.Note.Microsoft.VoiceMail.UM.CA
- IPM.Note.RPMSG.Microsoft.VoiceMail
- IPM.Note.RPMSG.Microsoft.VoiceMail.UM
- IPM.Note.RPMSG.Microsoft.VoiceMail.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.

2.2.2.56 UmUserNotes

The **UmUserNotes** element [<7>](#) 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.

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

- IPM.Note.Microsoft.VoiceMail
- IPM.Note.Microsoft.VoiceMail.UM
- IPM.Note.Microsoft.VoiceMail.UM.CA
- IPM.Note.RPMSG.Microsoft.VoiceMail
- IPM.Note.RPMSG.Microsoft.VoiceMail.UM
- IPM.Note.RPMSG.Microsoft.VoiceMail.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.

2.2.2.57 ConversationId

The **ConversationId** element [<8>](#) is a required byte-array that specifies a unique identifier for a conversation. It is defined as an element in the Email2 namespace.

The server fails conversation actions when the **ConversationId** is null.

The client MUST NOT change the **ConversationId** value.

2.2.2.58 ConversationIndex

The **ConversationIndex** element [<9>](#) is a required byte-array that contains a **ConversationId** and a set of dates and times. It is defined as an element in the Email2 namespace.

The **ConversationIndex** is used by clients to generate a conversation tree [view](#).

The **ABNF** format [\[RFC5322\]](#) of the **ConversationIndex** is as follows:

```
conversationindex = signature "+" 1stmsgtimestamp "+" GUID "+" [addtimestamps]
signature = "0x0E"
1stmsgtimestamp = timestamp
timestamp = date time
date = month "/" day "/" year
time = hour ":" minute ("AM"/"PM")
month = ([FWS]1*2DIGIT)
day = ([FWS]1*2DIGIT)
year = 4DIGIT
hour = 2DIGIT
minute = 2DIGIT
GUID = (16) DIGIT
addtimestamps = *timestamp
FWS =          ([*WSP CRLF] 1*WSP) / obs-FWS
              ; Folding white space
```

[<10>](#)

For example, a server can send a **message** with the following **ConversationIndex**, 0x0E+12/27/2006 11:51PM + {GUID}. If another user responded to the message at 12:56AM, the **ConversationIndex** of the response message would be 0x0E+12/27/2006 11:51PM + {GUID} + 1/5/2007 12:56AM.

To limit bandwidth, the server removes the 1 byte signature and the GUID from the **ConversationIndex** for every message, as that information is already available in the **ConversationId property**. The client MUST NOT change the **ConversationId** value.

2.2.2.59 LastVerbExecuted

The **LastVerbExecuted** element [<11>](#) 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

2.2.2.60 LastVerbExecutionTime

The **LastVerbExecutionTime** element [<12>](#) is an optional **datetime** element that indicates the time when the **LastVerbExecuted** (section [2.2.2.59](#)) was performed on the message. It is defined as an element in the Email2 namespace.

2.2.2.61 ReceivedAsBcc

The **ReceivedAsBcc** element [<13>](#) 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 2.2.1.19.2.19.

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

2.2.2.62 Sender

The **Sender** element [<14>](#) 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 2.2.1.19.2.19.

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.

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 [2.2.1.8](#), and section [2.2.2.40](#) through section [2.2.2.52](#).

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 [2.2.2.41](#).

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 2.2.1.19.2.19) 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.

A **reminder** cannot be added to a meeting request flag.

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.

Any of the complex types and elements in the E-mail class can be included in an **ItemOperations** command request.

E-mail class complex types and elements MUST be transmitted as children of the **Schema** type ([\[MS-ASCMD\]](#) section 2.2.1.8.2.12).

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.

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.

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

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 **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. The server **MUST** return an E-mail class **XML** block 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 Flagged 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

If the only elements with changes are the **Read** element or any of the **Flag** elements, then the server can send only those updates without sending the full E-mail class element again. If any other element of the e-mail message has changed, then the server MUST send the entire message to the client.

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

Read Flag	Flagging Properties	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

Read Flag	Flagging Properties	Other ActiveSync Protocol Properties	Non-ActiveSync Protocol Properties	Action
				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 2.2.1.8.3.8).

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 HTML E-Mail

4.1.1 Example Sync Request for Inbox with HTML Mail Support

The following example **Sync** command request synchronizes all mail in the Inbox, as specified by the **CollectionId** element.

```
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>
```

4.1.2 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>

```

4.1.3 Example Sync Response for E-Mail with One HTML Message

The following example shows the **Sync** command response from the server adding one e-mail item with an HTML body. The body, as well as metadata about its type and estimated size, is included within the **Body** node. In this case, the body has been truncated. To retrieve the complete body, the client sends the **ItemOperations** command to fetch the body content, as described in section [4.1.4](#).

Note that the response includes the **NativeBodyType** element (as specified in [\[MS-ASAIRS\]](#) section 2.2.2.10), which indicates that the message is stored as HTML on the server.

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
X-MS-MV: 14.0.255
Date: Thu, 19 Feb 2009 01:43:34 GMT
Content-Length: 399

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:Email="POOMMAIL:" 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>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>384</AirSyncBase:EstimatedDataSize>
              <AirSyncBase:Truncated>1</AirSyncBase:Truncated>
            </AirSyncBase:Body>
            <Email:MessageClass>IPM.Note</Email:MessageClass>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>

```



```

    <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>
    <Email:Categories />
  </ApplicationData>
</Add>
<Change>
  <ServerId>5:8</ServerId>
  <ApplicationData>
    <Email:Read>1</Email:Read>
  </ApplicationData>
</Change>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.4 Example Item Operations Request

The following example shows the **ItemOperations** command request to retrieve the truncated e-mail from section [4.1.3](#).

```

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>

```

4.1.5 Example Item Operations Response With Fetched Content

The following example shows the **ItemOperations** command response with the content that was truncated from the example in section [4.1.3](#). This message was sent in response to the request in section [4.1.4](#).

Note that, in the example code, **HTML** strings are escaped by using **<** and **>**. However, as these values are passed over the wire, they are passed in their original HTML format, as **<** and **>**.

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
X-MS-MV: 14.0.255
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;
          </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>
```

4.1.6 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
X-MS-MV: 14.0.255
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</ServerId>
          <ApplicationData>
            <Email:To>"Device User" &lt;deviceuser@example.com&gt;</Email:To>
            <Email:From>"7125550123" &lt;7125550123&gt;</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>

4.1.7 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
X-MS-MV: 14.0.511
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>
      <Commands>
        <Add>
          <ServerId>5:3</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>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>ÿgÈ:ConversationId>ÿgÈtent-clEmail2:ConversationId>
            <Email2:ConversationIndex>È:ConversatEmail2:ConversationIndex>
            <Email:Categories />
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

```
</Collection>
</Collections>
</Sync>
```

4.1.8 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 [4.1.3](#) 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>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Commands>
        <Delete>
          <ServerId>5:10</ServerId>
        </Delete>
      </Commands>
    </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
X-MS-MV: 14.0.255
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&gt;</Email:To>
  <Email:From>"Device User2" &lt;deviceuser2@example.com&gt;</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&gt;</Email:Organizer>
    <Email:Reminder>900</Email:Reminder>
    <Email:ResponseRequested>1</Email:ResponseRequested>
    <Email:Sensitivity>0</Email:Sensitivity>
    <Email:BusyStatus>2</Email:BusyStatus>

    <Email:TimeZone>aEAAACgARwBNAFQALQA wADYAOGAwADAAKQAgAEMAZQB uAHQAcgBhAGwAIA BUAGkAbQB lACA AKABVA
    FMAIAAAmCAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAAAAACgARwBNAFQALQA wADYAOGAwADAAKQAgAEMAZQB uAHQAcgBhAGwAIA
    BUAGkAbQB lACA AKABVAFMAIAAAmCAAQwAAAAMAAACAAIAAAAAAAAAAxP//w==</Email:TimeZone>

    <Email:GlobalObjId>BAAAAIAA4AB0xbCQGoLgCAAAAADYSxf9bJLJAQAAAAAAAAAAAAEAAAAJEHL7SUox5GtgQV1TYDY4
    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>Lððþ*û@â²&#x1A; &#x15;EñMø±</Email2:ConversationId>
    <Email2:ConversationIndex>É'ly&#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
X-MS-MV: 14.0.255
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>aAEAAcGARwBNAFQALQAwADYAOGwADAACKQAgAEMAZQBwAHQAAGwAIABUAGkAbQBlCAAKABVA
FMAIAAAmCAAQwAAAsAAAABAAIAAAAAAAAAAAAAACgARwBNAFQALQAwADYAOGwADAACKQAgAEMAZQBwAHQAAGwAIA
BUAGkAbQBlCAAKABVAFMAIAAAmCAAQwAAAMAAAAAAAIAAAAAAAAAAxP//w==</Email:TimeZone>
```

```

<Email:GlobalObjId>BAAAAIA4AB0xbcQGoLgCAAAAADok5WnbpLJAQAAAAAAAAAAAAEAAAAP4Ao5IYwQdKiFkDBeGTTg
Y=</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>'MÅ ' &amp; Kä °V=ŽÓ&#x16;xû</Email2:ConversationId>
  <Email2:ConversationIndex>É' n-,</Email2:ConversationIndex>
  <Email:Categories />
</ApplicationData>
</Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.1.11 Example Success Sync Response

The following example shows the **Sync** command response sent by the server after completing the deletion requested in section [4.1.8](#).

```

HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
X-MS-MV: 14.0.255
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>

```

4.2 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
X-MS-MV: 14.0.255
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:

- 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 the product does not follow the prescription.

[<1> Section 2.1:](#) Exchange 2007 does not support the Email2 namespace.

[<2> Section 2.2.2.12:](#) Exchange 2007 does not support the UmAttOrder element.

[<3> Section 2.2.2.13:](#) Exchange 2007 does not support the UmAttDuration element.

[<4> Section 2.2.2.14:](#) Exchange 2010 Beta does not support read receipts when client connection services are deployed on an Exchange server that does not also have a Mailbox store installed.

[<5> Section 2.2.2.38:](#) Exchange 2007 does not support the DisallowNewTimeProposal element.

[<6> Section 2.2.2.55:](#) Exchange 2007 does not support the UmCallerId element.

[<7> Section 2.2.2.56:](#) Exchange 2007 does not support the UmUserNotes element.

[<8> Section 2.2.2.57:](#) Exchange 2007 does not support the ConversationId element.

[<9> Section 2.2.2.58:](#) Exchange 2007 does not support the ConversationIndex element.

[<10> Section 2.2.2.58:](#) The 0x0E signature in the code sample indicates that the message was created using Exchange 2010.

[<11> Section 2.2.2.59:](#) Exchange 2007 does not support the LastVerbExecuted element.

[<12> Section 2.2.2.60:](#) Exchange 2007 does not support the LastVerbExecutionTime element.

[<13> Section 2.2.2.61:](#) Exchange 2007 does not support the ReceivedAsBcc element.

[<14> Section 2.2.2.62:](#) Exchange 2007 does not support the Sender element.

7 Change Tracking

This section will report content and/or editorial changes, beginning with the next release.

8 Index

C

[Change tracking](#)

E

[Examples - overview](#)

G

[Glossary](#)

I

[Introduction](#)

M

Messages
[overview](#)

N

[Normative references](#)

O

[Overview \(synopsis\)](#)

P

[Preconditions](#)
[Prerequisites](#)
[Product behavior](#)

R

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

S

Security
[overview](#)

T

[Tracking changes](#)