

# [MS-ASEMAIL]: ActiveSync E-Mail Class Protocol Specification

## Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. 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 protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting [protocol@microsoft.com](mailto:protocol@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

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

**Tools.** This protocol documentation is 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. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	December 3, 2008	1.0	Initial Release.
Microsoft Corporation	February 4, 2009	1.01	Revised and edited technical content.
Microsoft Corporation	March 4, 2009	1.02	Revised and edited technical content.

# Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Glossary .....	5
1.2	References .....	5
1.2.1	Normative References .....	5
1.2.2	Informative References .....	6
1.3	Protocol Overview .....	6
1.4	Relationship to Other Protocols.....	6
1.5	Prerequisites/Preconditions.....	6
1.6	Applicability Statement.....	6
1.7	Versioning and Localization.....	6
1.8	Vendor-Extensible Fields .....	6
1.9	Standards Assignments .....	6
<b>2</b>	<b>Messages.....</b>	<b>7</b>
2.1	Transport.....	7
2.2	Message Syntax.....	7
2.2.1	Complex Types.....	9
2.2.1.1	Attachments .....	10
2.2.1.2	Attachments.Attachment.....	10
2.2.1.3	Body .....	10
2.2.1.4	MeetingRequest.....	10
2.2.1.5	MeetingRequest.Recurrences .....	10
2.2.1.6	MeetingRequest.Recurrences.Recurrence.....	11
2.2.1.7	MeetingRequest.Categories .....	11
2.2.1.8	Flag.....	11
2.2.2	Elements.....	11
2.2.2.1	To .....	14
2.2.2.2	Cc .....	15
2.2.2.3	From .....	15
2.2.2.4	Subject.....	15
2.2.2.5	ReplyTo.....	15
2.2.2.6	DateReceived.....	15
2.2.2.7	DisplayTo.....	15
2.2.2.8	ThreadTopic.....	16
2.2.2.9	Importance .....	16
2.2.2.10	Read.....	16
2.2.2.11	Attachments.Attachment.DisplayName .....	16
2.2.2.12	MessageClass.....	16
2.2.2.13	MeetingRequest.AllDayEvent .....	17
2.2.2.14	MeetingRequest.StartTime .....	17
2.2.2.15	MeetingRequest.DtStamp .....	18
2.2.2.16	MeetingRequest.EndTime .....	18
2.2.2.17	MeetingRequest.InstanceType.....	18

2.2.2.18	MeetingRequest.Location .....	18
2.2.2.19	MeetingRequest.Organizer .....	18
2.2.2.20	MeetingRequest.RecurrenceId .....	18
2.2.2.21	MeetingRequest.Reminder .....	19
2.2.2.22	MeetingRequest.ResponseRequested .....	19
2.2.2.23	MeetingRequest.Recurrences.Recurrence.Type .....	19
2.2.2.24	MeetingRequest.Recurrences.Recurrence.Interval .....	19
2.2.2.25	MeetingRequest.Recurrences.Recurrence.Until .....	19
2.2.2.26	MeetingRequest.Recurrences.Recurrence.Occurrences .....	20
2.2.2.27	MeetingRequest.Recurrences.Recurrence.WeekOfMonth .....	20
2.2.2.28	MeetingRequest.Recurrences.Recurrence.DayOfMonth .....	20
2.2.2.29	MeetingRequest.Recurrences.Recurrence.DayOfWeek .....	20
2.2.2.30	MeetingRequest.Recurrences.Recurrence.MonthOfYear .....	21
2.2.2.31	MeetingRequest.Sensitivity .....	21
2.2.2.32	MeetingRequest.IntDBusyStatus .....	21
2.2.2.33	MeetingRequest.TimeZone .....	22
2.2.2.34	MeetingRequest.GlobalObjId .....	22
2.2.2.35	MeetingRequest.Categories.Category .....	22
2.2.2.36	InternetCPID .....	22
2.2.2.37	Flag.Subject .....	22
2.2.2.38	Flag.Status .....	22
2.2.2.39	Flag.FlagType .....	23
2.2.2.40	Flag.DateCompleted .....	23
2.2.2.41	Flag.CompleteTime .....	23
2.2.2.42	Flag.StartDate .....	23
2.2.2.43	Flag.DueDate .....	23
2.2.2.44	Flag.UTCStartDate .....	24
2.2.2.45	Flag.UTCEndDate .....	24
2.2.2.46	Flag.ReminderSet .....	24
2.2.2.47	Flag.ReminderTime .....	24
2.2.2.48	Flag.OrdinalDate .....	24
2.2.2.49	Flag.SubOrdinalDate .....	24
2.2.2.50	ContentClass .....	25
2.2.2.51	NativeBodyType .....	25
<b>3</b>	<b>Protocol Details.....</b>	<b>25</b>
3.1	Client Details .....	25
3.1.1	Abstract Data Model .....	25
3.1.2	Timers .....	25
3.1.3	Initialization .....	25
3.1.4	Higher-Layer Triggered Events.....	25
3.1.4.1	Synchronizing E-mail Between Client and Server .....	25
3.1.4.2	Searching E-mail .....	26
3.1.4.3	Retrieving Individual E-mail .....	26
3.1.4.4	Sending Flag Updates to the Server .....	26

3.1.5	Message Processing Events and Sequencing Rules .....	27
3.1.5.1	ItemOperations Command Request .....	27
3.1.5.2	Search Command Request .....	28
3.1.5.3	Sync Command Request .....	28
3.1.6	Timer Events.....	28
3.1.7	Other Local Events.....	28
3.2	Server Details .....	28
3.2.1	Abstract Data Model .....	28
3.2.2	Timers .....	29
3.2.3	Initialization .....	29
3.2.4	Higher-Layer Triggered Events.....	29
3.2.4.1	Synchronizing E-mail Between Client and Server .....	29
3.2.4.2	Searching E-mail .....	29
3.2.4.3	Retrieving Individual E-mail .....	29
3.2.4.4	Change Tracking Semantics for Flagging Properties .....	29
3.2.4.5	Sending Flagged Changes to the Client.....	30
3.2.5	Message Processing Events and Sequencing Rules .....	31
3.2.5.1	ItemOperations Command Response.....	31
3.2.5.2	Search Command Response .....	31
3.2.5.3	Sync Command Response .....	32
3.2.6	Timer Events.....	32
3.2.7	Other Local Events.....	32
<b>4</b>	<b><i>Protocol Examples</i></b> .....	<b>32</b>
4.1	Synchronizing HTML E-Mail .....	32
4.1.1	Sample Sync Request for Inbox with HTML Mail Support .....	32
4.1.2	Sample Sync Request for Inbox with Body Preferences.....	32
4.1.3	Sample Sync Response for E-Mail with One HTML Message .....	33
4.2	Setting Flags on the Client and Server .....	34
4.2.1	Setting a Flag on the Client.....	35
4.2.2	Setting a Flag on the Server .....	36
4.2.3	Setting the Complete Flag.....	36
4.2.4	Clearing a Flag on the Client .....	37
<b>5</b>	<b><i>Security</i></b> .....	<b>38</b>
5.1	Security Considerations for Implementers.....	38
5.2	Index of Security Parameters.....	38
<b>6</b>	<b><i>Appendix A: Office/Exchange Behavior</i></b> .....	<b>38</b>
	<b><i>Index</i></b> .....	<b>39</b>

# 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**  
**collection**  
**Coordinated Universal Time (UTC)**  
**message database (MDB)**  
**WAP Binary XML (WBXML)**

The following terms are specific to this document:

**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

[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-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>.

[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, and is 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**

None.

## **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 two namespaces: Email 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], 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: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:element name="EndTime" type="xs:dateTime" />
        <xs:element name="InstanceType"
type="xs:unsignedByte" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <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:unsignedByte" />
                            <xs:element
name="Until" type="xs:string" />
                            <xs:element
name="Occurrences" type="xs:unsignedByte" />
                            <xs:element
name="WeekOfMonth" type="xs:unsignedByte" />
                            <xs:element
name="DayOfMonth" type="xs:unsignedByte" />
                            <xs:element
name="DayOfWeek" type="xs:unsignedByte" />
                            <xs:element
name="MonthOfYear" type="xs:unsignedByte" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Sensitivity"
type="xs:unsignedByte" />
    <xs:element name="IntDBusyStatus"
type="xs:unsignedByte" />
    <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="unbounded"
name="Category" type="xs:string" />
            </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:unsignedByte" />
      <xs:element name="FlagType" type="xs:string" />
      <xs:element name="DateCompleted" type="xs:dateTime" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="CompleteTime" type="xs:dateTime" />
<xs:element name="StartDate" type="xs:dateTime" />
<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>

```

## 2.2.1 Complex Types

The following table summarizes the set of common XML schema complex type definitions 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.Categories	A collection of <b>Category</b> elements belonging to a <b>MeetingRequest</b> .
MeetingRequest.Recurrences	A collection of <b>Recurrence</b> elements.
MeetingRequest.Recurrences.Recurrence	A collection of <b>Recurrence</b> elements that describe when and how often this meeting recurs.

Flag	The flag associated with the item, along with its 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.

The **Attachments** type is part of the AirSyncBase namespace, and is further specified in [MS-ASAIRS] section 2.2.3.1.

### 2.2.1.2 Attachments.Attachment

The **Attachment** type is a **container** ([MS-ASDTYPE] section 2.8) type that represents an e-mail attachment.

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

The **Attachment** type is part of the AirSyncBase namespace, and is further specified in [MS-ASAIRS] section 2.2.3.2.

### 2.2.1.3 Body

The **Body** type is an optional **container** ([MS-ASDTYPE] section 2.8) 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 2.2.3.3.

### 2.2.1.4 MeetingRequest

The **MeetingRequest** type is an optional **container** ([MS-ASDTYPE] section 2.8) type that signifies that this e-mail meeting corresponds to a meeting request.

### 2.2.1.5 MeetingRequest.Recurrences

The **MeetingRequest.Recurrences** type is a **container** ([MS-ASDTYPE] section 2.8) 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** ([MS-ASDTYPE] section 2.8) 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** ([MS-ASDTYPE] section 2.8) 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 ([MS-ASDTYPE] section 2.8) type that describes the flag associated with this item, along with its current status.

## 2.2.2 Elements

The following table summarizes 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 the operation.

E-mail class elements **MUST NOT** have child elements in the command request or response.

Element	Description
<b>To</b>	The list of recipients.
<b>Cc</b>	The list of carbon-copy recipients.
<b>From</b>	The e-mail address of the individual who sent the message.
<b>Subject</b>	The subject of the e-mail message.
<b>ReplyTo</b>	The e-mail address to which replies will be addressed by default.
<b>DateReceived</b>	The date and time that the message was received on the server.

<b>DisplayTo</b>	The names of the primary recipients of the message.
<b>ThreadTopic</b>	The topic used in conversation reading.
<b>Importance</b>	The importance of the message, as determined by the sender.
<b>Read</b>	Specifies whether the message has been read.
<b>Attachments.Attachment.DisplayName</b>	The name of the attachment file as displayed to the user.
<b>MessageClass</b>	The message <b>class</b> of this e-mail message.
<b>MeetingRequest.AllDayEvent</b>	Indicates whether the calendar item is an all day event.
<b>MeetingRequest.StartTime</b>	The date and time that the <b>MeetingRequest</b> element starts.
<b>MeetingRequest.DtStamp</b>	The date and time that the calendar item was created.
<b>MeetingRequest.EndTime</b>	The date and time that the <b>MeetingRequest</b> element ends.
<b>MeetingRequest.InstanceType</b>	The type of calendar item.
<b>MeetingRequest.Location</b>	The location for the calendar item.
<b>MeetingRequest.Organizer</b>	The SMTP e-mail alias of the meeting organizer.
<b>MeetingRequest.RecurrenceId</b>	A specific instance of a recurring calendar item.

<b>MeetingRequest.Reminder</b>	The number of seconds prior to the calendar item's start time that a reminder is displayed.
<b>MeetingRequest.ResponseRequested</b>	Indicates whether the originator of the meeting has requested a response.
<b>MeetingRequest.Recurrences.Recurrence.Type</b>	The recurrence type.
<b>MeetingRequest.Recurrences.Recurrence.Interval</b>	The interval between recurrences.
<b>MeetingRequest.Recurrences.Recurrence.Until</b>	The end time of a series of recurrence items.
<b>MeetingRequest.Recurrences.Recurrence.Occurrences</b>	The number of occurrences before the series ends.
<b>MeetingRequest.Recurrences.Recurrence.WeekOfMonth</b>	The week of the month.
<b>MeetingRequest.Recurrences.Recurrence.DayOfMonth</b>	The day of the month.
<b>MeetingRequest.Recurrences.Recurrence.DayOfWeek</b>	The day of the week.
<b>MeetingRequest.Recurrences.Recurrence.MonthOfYear</b>	The month of the year.
<b>MeetingRequest.Sensitivity</b>	The sensitivity level of the meeting request.
<b>MeetingRequest.IntDBusyStatus</b>	The intended busy status for the meeting request.
<b>MeetingRequest.TimeZone</b>	The time zone where the calendar item occurs.
<b>MeetingRequest.GlobalObjId</b>	The Base64-encoded global object ID for the meeting request.
<b>MeetingRequest.Categories.Category</b>	A named label for the <b>MeetingRequest</b> element.
<b>InternetCPID</b>	The original code page ID from the MIME message.

<b>Flag.Subject</b>	The subject of the flag as it would appear in a task list.
<b>Flag.Status</b>	The current status of the flag.
<b>Flag.FlagType</b>	The value of the <b>Flag To: follow up</b> field.
<b>Flag.DateCompleted</b>	The date on which the flagged item was completed.
<b>Flag.CompleteTime</b>	The time at which the flagged item was marked as finished.
<b>Flag.StartDate</b>	The start date of the flagged item.
<b>Flag.DueDate</b>	The due date of the flagged item.
<b>Flag.UTCStartDate</b>	The <b>Coordinated Universal Time (UTC)</b> value of the local <b>StartDate</b> .
<b>Flag.UTCEndDate</b>	The <b>UTC</b> value of the local <b>DueDate</b> .
<b>Flag.ReminderSet</b>	Identifies whether a reminder has been set for this flagged item.
<b>Flag.ReminderTime</b>	The date and time that the reminder is supposed to occur.
<b>Flag.OrdinalDate</b>	The time at which the client set the flag.
<b>Flag.SubOrdinalDate</b>	A string used to sort items.
<b>NativeBodyType</b>	The format in which the item is stored on the server.
<b>ContentClass</b>	The content <b>class</b> of the data.

### 2.2.2.1 To

The **To** element is an optional element that specifies the list of recipients.

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.

#### **2.2.2.2 Cc**

The **Cc** element is an optional element that specifies the list of carbon-copied recipients of this 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.

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.

#### **2.2.2.4 Subject**

The **Subject** element is an optional element that specifies the subject of the e-mail message.

#### **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.

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.

#### **2.2.2.6 DateReceived**

The **DateReceived** element is an optional element that specifies the date and time when this message was received by the current recipient.

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

#### **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.

### 2.2.2.8 ThreadTopic

The **ThreadTopic** element is an optional element that specifies the topic used for conversation threading.

### 2.2.2.9 Importance

The **Importance** element is an optional element that specifies the importance of the message, as determined by the sender.

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.

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

### 2.2.2.11 Attachments.Attachment.DisplayName

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

### 2.2.2.12 MessageClass

The **MessageClass** element is an optional element that specifies the message **class** of this e-mail message.

The value of the **MessageClass** element **SHOULD** be one of the following values.

Value	Meaning
IPM.Note	Normal e-mail message
IPM.Note.Rules.OofTemplate.Microsoft	Out-of-office message
IPM.Note.SMIME	Secure MIME (S/MIME) encrypted and opaque-signed message



IPM.Note.SMIME.MultipartSigned	S/MIME clear-signed message
IPM.Schedule.Meeting.Request	Message containing a meeting request
IPM.Schedule.Meeting.Canceled	Notification of a canceled meeting
IPM.Schedule.Meeting.Resp.Pos	Accepted meeting request
IPM.Schedule.Meeting.Resp.Tent	Tentatively accepted meeting request
IPM.Schedule.Meeting.Resp.Neg	Declined meeting request
IPM.Post	Post

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.

<b>MessageClass</b>	<b>Description</b>
REPORT.<message class>.DR	Deliver receipt
REPORT.<message class>.NDR	Non-delivery receipt
REPORT.<message class>.IPNRN	Message read report
REPORT.<message class>.IPNNRN	Message not read report

For example, a standard message non-delivery receipt is **REPORT.IPM.Note.NDR** and a delivery receipt for a meeting is **REPORT.IPM.Schedule.Meeting.Request.DR**.

### 2.2.2.13 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.

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.

### 2.2.2.14 MeetingRequest.StartTime

The **MeetingRequest.StartTime** element is a required child element of the **MeetingRequest** type that specifies when this meeting begins.

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

### 2.2.2.15 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.

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

### 2.2.2.16 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.

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

### 2.2.2.17 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.

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

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

### 2.2.2.18 MeetingRequest.Location

The **MeetingRequest.Location** element is an optional child element of the **MeetingRequest** type that specifies where this meeting will be held.

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

### 2.2.2.19 MeetingRequest.Organizer

The **MeetingRequest.Organizer** element is an optional child element of the **MeetingRequest** type that specifies who organized this meeting.

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

### 2.2.2.20 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.

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.

#### **2.2.2.21 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.

#### **2.2.2.22 MeetingRequest.ResponseRequested**

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

#### **2.2.2.23 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.

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

<b>Value</b>	<b>Meaning</b>
0	Rekurs daily.
1	Rekurs weekly.
2	Rekurs monthly.
3	Rekurs monthly on the nth day of the month.
5	Rekurs yearly.
6	Rekurs yearly on the nth day of the year.

#### **2.2.2.24 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.

#### **2.2.2.25 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.

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

### 2.2.2.26 MeetingRequest.Recurrences.Recurrence.Occurrences

The **MeetingRequest.Recurrences.Recurrence.Occurrences** element is a required child element of the **Recurrence** type that specifies the number of occurrences before the series ends.

### 2.2.2.27 MeetingRequest.Recurrences.Recurrence.WeekOfMonth

The **MeetingRequest.Recurrences.Recurrence.DayOfMonth** element is an optional child element of the **Recurrence** type that specifies the week of the month in which this meeting recurs.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 5.

### 2.2.2.28 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.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 3 or 6.

### 2.2.2.29 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.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 1, 2 or 6.

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.

### 2.2.2.30 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.

This element is required when the **MeetingRequest.Recurrences.Recurrence.Type** element is set to 6.

### 2.2.2.31 MeetingRequest.Sensitivity

The **MeetingRequest.Sensitivity** element is an optional child element of the **MeetingRequest** type that specifies the sensitivity level of the meeting request.

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.

### 2.2.2.32 MeetingRequest.IntDBusyStatus

The **MeetingRequest.IntDBusyStatus** 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.

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.

### **2.2.2.33 MeetingRequest.TimeZone**

The **MeetingRequest.TimeZone** element is a required child element of the **MeetingRequest** type that specifies the time zone where the calendar item is recurring.

The value of this element is a **timezone** value, as specified in [MS-ASDTYPE] section 2.7.

### **2.2.2.34 MeetingRequest.GlobalObjId**

The **MeetingRequest.GlobalObjId** element is a required child element of the **MeetingRequest** type that specifies the base64-encoded global object ID for the meeting request.

### **2.2.2.35 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.

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.

### **2.2.2.36 InternetCPID**

The **InternetCPID** element is a required element that contains the original code page ID from the MIME message.

### **2.2.2.37 Flag.Subject**

The **Flag.Subject** element is an optional child element of the **Flag** type that specifies the subject of the flag.

The client or server **SHOULD** set the value of this element to the subject of the message when an item is flagged.<1>

### **2.2.2.38 Flag.Status**

The **Flag.Status** element is an optional child element of the **Flag** type that specifies the current status of the flag.

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

### 2.2.2.39 Flag.FlagType

The **Flag.FlagType** element is an optional child element of the **Flag** type that specifies the flag type.

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

New flags MUST always set this field to “Follow up”.

### 2.2.2.40 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.

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

### 2.2.2.41 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.

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

### 2.2.2.42 Flag.StartDate

The **Flag.StartDate** element is an optional child element of the **Flag** type that specifies when this flagged item was begun.

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

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

### 2.2.2.43 Flag.DueDate

The **Flag.DueDate** element is an optional child element of the **Flag** type that specifies when this flagged item is due.

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

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

#### **2.2.2.44 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**.

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

#### **2.2.2.45 Flag.UTCEndDate**

The **Flag.UTCEndDate** element is an optional child element of the **Flag** type that contains the **UTC** value of local **Flag.DueDate**.

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

#### **2.2.2.46 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).

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

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

#### **2.2.2.47 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 supposed to occur.

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

#### **2.2.2.48 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.<2>

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

#### **2.2.2.49 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**.<3>

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



### 2.2.2.50 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”.

### 2.2.2.51 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 2.2.4.11.

## 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 Synchronizing 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 Searching 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 Retrieving Individual E-mail

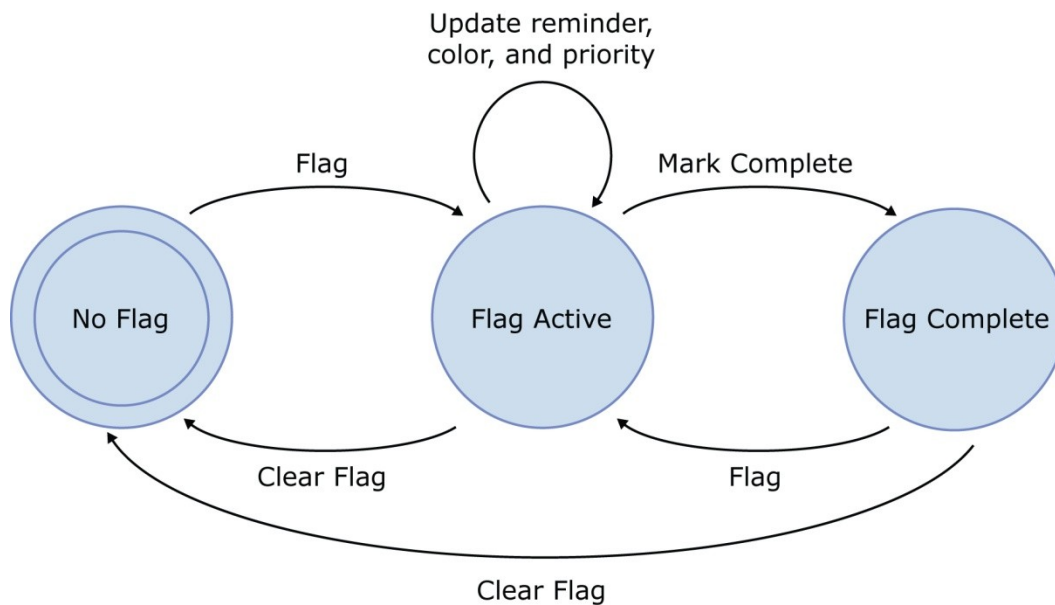
E-mail data for one or more individual e-mail items is requested by the client using an **ItemOperations** command request, which is a wrapper for the **Fetch** command. An **ItemOperations** command can contain multiple **Fetch** commands. The **ItemOperations** command request is specified in section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

### 3.1.4.4 Sending 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 sections 2.2.1.8, and 2.2.2.37 through 2.2.2.49.

The following figure shows the life cycle of a flag.

**Figure 1: Flag life cycle**



Task integration with flagging is optional for the client, and layers cleanly on top of the simple 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 2.2.2.38.

Action	Required Properties from Device
Flag an item (basic)	<b>Status</b> = 2 <b>FlagType</b> = 'for follow-up' <b>StartDate</b> and <b>UTCStartDate</b> <b>DueDate</b> and <b>UTCDueDate</b>
Flag an item (task flagging)	<b>Status</b> = 2 <b>Subject</b> = <user defined> <b>FlagType</b> = follow-up <b>StartDate</b> and <b>UTCStartDate</b> <b>DueDate</b> and <b>UTCDueDate</b> <b>ReminderSet</b> <b>ReminderTime</b>
Mark an item complete (basic)	<b>Status</b> = 1 <b>CompleteTime</b> <b>DateCompleted</b>
Mark an item complete (task flagging)	<b>Status</b> = 1 <b>CompleteTime</b> <b>DateCompleted</b>
Clearing the flag on an item	Flag node empty
Clearing the flag on an item (task flagging)	Flag node empty
Update the flag metadata (basic)	All updated properties
Update flag metadata (task flagging)	All updated properties

If neither **StartDate**, **DueDate**, **UtcStartDate**, nor **UtcDueDate** exist in the request, then it is valid to accept O11 flag.

**DateCompleted** is required to accept O11 flag.

**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 2.2.1.19.1.6.

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 Synchronizing 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 section 3.2.5.3 and [MS-ASCMD] section 2.2.1.19.2.

#### 3.2.4.2 Searching 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 3.1.5.2 and [MS-ASCMD] section 2.2.1.14.2.

#### 3.2.4.3 Retrieving 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 3.1.5.1 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.

### 3.2.4.5 Sending Flagged Changes to the Client

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

- 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.

<b>Read Flag</b>	<b>Flagging Properties</b>	<b>Other ActiveSync Protocol Properties</b>	<b>Non-ActiveSync Protocol Properties</b>	<b>Action</b>
N	N	N	N	Send nothing to client
N	N	N	Y	Send nothing to client
N	N	Y	N	Send full item <b>Change</b> to client
N	N	Y	Y	Send full item <b>Change</b> 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 <b>Change</b> to client
N	Y	Y	Y	Send full item <b>Change</b> to client
Y	N	N	N	Send read flag only
Y	N	N	Y	Send read flag

Read Flag	Flagging Properties	Other ActiveSync Protocol Properties	Non-ActiveSync Protocol Properties	Action
				only
Y	N	Y	N	Send full item <b>Change</b> to client
Y	N	Y	Y	Send full item <b>Change</b> 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 <b>Change</b> 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 2.2.1.21.1.6.

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

### 4.1 Synchronizing HTML E-Mail

#### 4.1.1 Sample Sync Request for Inbox with HTML Mail Support

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

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v121Device&DeviceType=SmartPhone
Content-Type: application/vnd.ms-sync.wbxml
X-MS-PolicyKey: 230840124
MS-ASProtocolVersion: 12.1

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>646483458</SyncKey>
      <CollectionId>8</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
    </Collection>
  </Collections>
</Sync>
```

#### 4.1.2 Sample 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** and **AllOrNone** values for each **Type** value. For more information about the **BodyPreference**, **TruncationSize**, **AllOrNone**, or **Type** elements, see [MS-ASAIRS] section 2.2.

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v121Device
&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 12.1

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>646483458</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves/>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Options>
        <FilterType>2</FilterType>
        <AirSyncBase:BodyPreference>
          <AirSyncBase:Type>1</AirSyncBase:Type>
          <AirSyncBase:TruncationSize>5120</AirSyncBase:TruncationSize>
          <AirSyncBase:AllOrNone>1</AirSyncBase:AllOrNone>
        </AirSyncBase:BodyPreference>
        <AirSyncBase:BodyPreference>
          <AirSyncBase:Type>2</AirSyncBase:Type>
          <AirSyncBase:TruncationSize>5120</AirSyncBase:TruncationSize>
          <AirSyncBase:AllOrNone>1</AirSyncBase:AllOrNone>
        </AirSyncBase:BodyPreference>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

### 4.1.3 Sample Sync Response for E-Mail with One HTML Message

The following example response shows the server returning 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 not been truncated. If it had been truncated, then the **Body** node would also include a flag to indicate that the message has been truncated.

Note that the response includes the **NativeBodyType** element (as specified in [MS-ASAIRS] section 2.2.4.11), which indicates that the message is stored as HTML on the server. Also note that, in the example code, HTML strings are escaped by using **&lt;** and **&gt;**. 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-Length: 347
Content-Type: application/vnd.ms-sync.wbxml

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1601897837</SyncKey>
      <CollectionId>7</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:To>"deviceuser" &lt;someone@example.com&gt;
</email:To>
            <email:From>"deviceuser2" &lt;someone2@example.com&gt;
</email:From>
            <email:Subject>HTML EMAIL</email:Subject>
            <email:DateReceived>2007-05-08T17:57:22.890Z
</email:DateReceived>
            <email:DisplayTo>deviceuser</email:DisplayTo>
            <email:ThreadTopic>HTML EMAIL</email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>0</email:Read>
            <airsyncbase:Body>
              <airsyncbase:Type>2</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>58
</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
</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>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>

```

## 4.2 *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.

#### 4.2.1 Setting a Flag on the Client

The following is an example of how to set a flag in a request message.

```
POST /Microsoft-ServerActiveSync?Cmd=Sync&User=
deviceuser&DeviceId=v121Device&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 12.1

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:" xmlns:tasks="POOMTASKS:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1601897839</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Commands>
        <Change>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:Read>1</email:Read>
            <email:Flag>
              <tasks:DueDate>2009-11-09T00:00:00.000Z</tasks:DueDate>
              <tasks:UtcDueDate>2009-11-09T08:00:00.000Z
              </tasks:UtcDueDate>
              <tasks:UtcStartDate>2009-11-02T08:00:00.000Z
              </tasks:UtcStartDate>
              <tasks:SUBOrdinalDate>555555</tasks:SUBOrdinalDate>
              <tasks:OrdinalDate>2009-11-02T08:30:00.000Z
              </tasks:OrdinalDate>
              <tasks:Subject>Flag Subject</tasks:Subject>
              <email:Status>2</email:Status>
              <email:FlagType>Follow up</email:FlagType>
              <tasks:StartDate>2009-11-02T00:00:00.000Z
              </tasks:StartDate>
              <tasks:ReminderSet>1</tasks:ReminderSet>
              <tasks:ReminderTime>2009-11-10T00:00:00.000Z
              </tasks:ReminderTime>
            </email:Flag>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

## 4.2.2 Setting a Flag on the Server

The following is an example response message showing a flag being sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>;
<Sync xmlns:email="POOMMAIL:" xmlns:tasks="POOMTASKS:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <Collection>
        <SyncKey>1588335828</SyncKey>
        <CollectionId>7</CollectionId>
        <Status>1</Status>
        <Commands>
          <Change>
            <ServerId>7:2</ServerId>
            <ApplicationData>
              <email:Flag>
                <tasks:DueDate>2009-11-09T00:00:00.000Z</tasks:DueDate>
                <tasks:UtcDueDate>2009-11-09T08:00:00.000Z
                </tasks:UtcDueDate>
                <tasks:UtcStartDate>2009-11-02T08:00:00.000Z
                </tasks:UtcStartDate>
                <tasks:SUBOrdinalDate>5555555</tasks:SUBOrdinalDate>
                <email:Status>2</email:Status>
                <email:FlagType>Test Follow up</email:FlagType>
                <tasks:StartDate>2009-11-02T00:00:00.000Z
                </tasks:StartDate>
                <tasks:ReminderSet>1</tasks:ReminderSet>
                <tasks:ReminderTime>2009-11-10T00:00:00.000Z
                </tasks:ReminderTime>
                <tasks:Subject>Flag Subject</tasks:Subject>
                <tasks:OrdinalDate>2009-11-02T08:30:00.000Z
                </tasks:OrdinalDate>
              </email:Flag>
            </ApplicationData>
          </Change>
        </Commands>
      </Collection>
    </Collections>
  </Sync>;
```

## 4.2.3 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 `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=v121Device&DeviceType=
```

```

PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 12.1

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:" xmlns:tasks="POOMTASKS" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1648049407</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves />
      <GetChanges />
      <Commands>
        <Change>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:Flag>
              <email:Status>1</email:Status>
              <tasks:DateCompleted>2007-05-08T10:24:26.765Z
            </tasks:DateCompleted>
              <email:CompleteTime>2007-05-08T10:24:26.765Z
            </email:CompleteTime>
            </email:Flag>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>

```

#### 4.2.4 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=v121Device&DeviceType=
PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 12.1

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:" xmlns:tasks="POOMTASKS:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1648049411</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Commands>
        <Change>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:Flag/>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>

```

```
</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: Office/Exchange Behavior

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

- Office 2007 with Service Pack 1 applied
- Exchange 2007 with Service Pack 1 applied

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

---

<1> Section 2.2.2.37: This element contains the subject that appears in the **To-Do Bar** in Office Outlook 2007. The only time this string differs from the subject of the message is when the user changes the task that appears in the **To-Do Bar**.

<2> Section 2.2.2.48: The **Flag.OrdinalDate** element is used for sorting items within the **To-Do Bar** in Office Outlook 2007.

<3> Section 2.2.2.49: The **Flag.SubOrdinalDate** element is used for sorting items within the **To-Do Bar** in Office Outlook 2007.

## Index

- Applicability statement, 6
- Introduction, 5
  - Glossary, 5
  - Protocol Overview, 6
  - References, 5
  - Relationship to Other Protocols, 6
- Messages, 7
- Office/Exchange Behavior, 38
- Prerequisites/preconditions, 6
- Protocol Details, 25
- Protocol Examples, 32
- Security, 38
- Standards assignments, 6
- Vendor-extensible fields, 6
- Versioning and localization, 6