

[MS-ASAIRS]: ActiveSync AirSyncBase Namespace 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.
- **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.

Table of Contents

1	Introduction.....	4
1.1	Glossary	4
1.2	References	4
1.2.1	Normative References	4
1.2.2	Informative References	4
1.3	Protocol Overview	5
1.4	Relationship to Other Protocols.....	5
1.5	Prerequisites/Preconditions.....	5
1.6	Applicability Statement.....	5
1.7	Versioning and Capability Negotiation.....	5
1.8	Vendor-Extensible Fields	5
1.9	Standards Assignments	5
2	Messages.....	6
2.1	Transport.....	6
2.2	Message Syntax.....	6
2.2.1	Namespaces	8
2.2.2	Simple Types	8
2.2.3	Complex Types.....	8
2.2.3.1	Attachments	8
2.2.3.2	Attachment.....	9
2.2.3.3	Body	9
2.2.3.4	BodyPreference	9
2.2.4	Elements.....	10
2.2.4.1	AllOrNone	11
2.2.4.2	ContentId.....	12
2.2.4.3	ContentLocation	12
2.2.4.4	ContentType	13
2.2.4.5	Data	13
2.2.4.6	DisplayName	13
2.2.4.7	EstimatedDataSize.....	13
2.2.4.8	FileReference.....	14
2.2.4.9	IsInline.....	14
2.2.4.10	Method	14
2.2.4.11	NativeBodyType	15
2.2.4.12	Truncated	16
2.2.4.13	TruncationSize.....	16
2.2.4.14	Type.....	16
2.2.5	Attributes.....	17
2.2.6	Groups.....	17
2.2.6.1	TopLevelSchemaProps	17
2.2.7	Attribute Groups.....	17

3	<i>Protocol Details</i>	18
3.1	Client Details	18
3.1.1	Abstract Data Model	18
3.1.2	Timers	18
3.1.3	Initialization	18
3.1.4	Higher-Layer Triggered Events.....	18
3.1.5	Message Processing Events and Sequencing Rules	18
3.1.5.1	Commands	18
3.1.5.1.1	Fetch	18
3.1.5.1.2	Search.....	18
3.1.5.1.3	Sync.....	19
3.1.6	Timer Events.....	19
3.1.7	Other Local Events.....	19
3.2	Server Details	19
3.2.1	Abstract Data Model	19
3.2.2	Timers	19
3.2.3	Initialization	19
3.2.4	Higher-Layer Triggered Events.....	19
3.2.5	Message Processing Events and Sequencing Rules	19
3.2.5.1	Commands	19
3.2.5.1.1	Sync.....	20
4	<i>Protocol Examples</i>	20
5	<i>Security</i>	20
5.1	Security Considerations for Implementers.....	20
5.2	Index of Security Parameters.....	20
6	<i>Appendix A: Office/Exchange Behavior</i>	20
	<i>Index</i>	21

1 Introduction

This document specifies the elements and complex types in the AirSyncBase namespace, which are used by the AirSync commands specified in [MS-ASCMD] to identify the size, type, and content of the data returned to the **client** in the response message. The AirSyncBase namespace contains elements and complex types used in both request and response command messages.

1.1 Glossary

The following terms are defined in [MS-OXGLOS]:

- client**
- collection**
- server**
- XML**
- XML schema definition (XSD)**

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-ASCMD] Microsoft Corporation, "ActiveSync Command Reference Protocol Specification", December 2008.

[MS-ASDTYPE] Microsoft Corporation, "ActiveSync Data Type Protocol Specification", December 2008.

[MS-ASEMAIL] Microsoft Corporation, "ActiveSync E-Mail Class 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>.

1.2.2 Informative References

None.

1.3 Protocol Overview

The elements and complex types specified in the AirSyncBase namespace are used by multiple Exchange ActiveSync commands to identify the size, type, and content of data returned to the **client** in the response message. In order to use the elements in the AirSyncBase namespace, the namespace is included in the command response message, and then elements and complex types are included in the request and response as specified in this document.

1.4 Relationship to Other Protocols

The **Fetch**, **Search**, and **Sync** commands use properties from the AirSyncBase namespace as a part of their request and response commands. For more details about these commands, see [MS-ASCMD].

1.5 Prerequisites/Preconditions

Before a command can use the properties in the AirSyncBase namespace, it must include the namespace in the command request. The namespace is included by adding the following to the command request:

```
<CommandName xmlns:airsyncbase="ClassName:">
```

For a complete example, see [MS-ASCMD] section 4.1.1.

1.6 Applicability Statement

This specification applies to the **Fetch**, **Search**, and **Sync** commands, as specified in [MS-ASCMD].

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

The **Type** element can be extended to include custom message types. For more details, see section 2.2.4.14.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The properties specified in the following sections are sent and received by using the **Fetch**, **Search**, and **Sync** commands, as specified in [MS-ASCMD], and are used by the E-mail class, as specified in [MS-ASEMAIL].

2.2 Message Syntax

The AirSyncBase namespace adheres to the following **XML schema definition (XSD)**:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified"
xmlns:airsyncbase="AirSyncBase:" elementFormDefault="qualified"
targetNamespace="AirSyncBase:"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="AirSyncBase:">
<xs:element name="FileReference" type="xs:string" />
<xs:element name="BodyPreference">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Type" type="xs:unsignedByte" />
      <xs:element name="TruncationSize"
minOccurs="0" type="xs:unsignedInt"/>
      <xs:element name="AllOrNone" minOccurs="0" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Body">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Type" type="xs:unsignedByte" />
      <xs:element name="EstimatedDataSize" type="xs:unsignedInt"
minOccurs="0"/>
      <xs:element name="Truncated" minOccurs="0" type="xs:boolean"/>
      <xs:element name="Data" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

```

    </xs:complexType>
</xs:element>
<xs:element name="Attachments">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Attachment">
        <xs:complexType>
          <xs:all>
<xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
          <xs:element name="FileReference" type="xs:string" />
          <xs:element name="Method" type="xs:unsignedByte" />
          <xs:element name="EstimatedDataSize"
type="xs:unsignedInt" />
          <xs:element name="ContentId" type="xs:string"
minOccurs="0"/>
          <xs:element name="ContentLocation" type="xs:string"
minOccurs="0"/>
          <xs:element name="IsInline" minOccurs="0"
type="xs:boolean"/>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="EmptyTag"/>
<xs:element name="NativeBodyType" type="xs:unsignedByte" />
<xs:group name="TopLevelSchemaProps">
  <xs:sequence>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="Body" type="airsynbase:EmptyTag"/>
      <xs:element name="Attachments" type="airsynbase:EmptyTag"/>
    </xs:choice>
  </xs:sequence>

```

```
</xs:group>
</xs:schema>
```

2.2.1 Namespaces

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

Prefix	Namespace URI
AIRSYNCBASE:	None.

2.2.2 Simple Types

This specification does not define any common **XML** schema simple type definitions.

2.2.3 Complex Types

The following table summarizes the set of common **XML** schema complex type definitions defined by this specification. XML schema complex type definitions that are specific to a particular operation are described with the operation.

Complex Type	Description
Body	A collection of elements that contain the body of the calendar, contact, document, email, or task.
BodyPreference	A collection of elements that sets the preference information related to the type and size of information that is returned from searching, synchronizing, or fetching.

2.2.3.1 Attachments

The **Attachments** type is an optional child element of the **Properties** element in the **Fetch** command ([MS-ASCMD] section 2.2.1.9.3.2) that contains one or more attachment items.

There **MUST** be no limit on the number of **Attachments** types in a command response. Command requests **MUST NOT** include the **Attachments** type.

The **Attachment** type **MUST** have at least one child **Attachment** type.

2.2.3.2 Attachment

The **Attachment** type is a required child element of the **Attachments** type and specifies the attachment information for a single attachment item.

Command requests MUST NOT include the **Attachment** type.

The **Attachment** type can have the one or more of the following child elements in any order (elements noted as optional MAY be in the response):

- **DisplayName** (section 2.2.4.6). This element is optional.
- **FileReference** (section 2.2.4.8). This element is required.
- **Method** (section 2.2.4.10). This element is required.
- **EstimatedDataSize** (section 2.2.4.7). This element is required.
- **ContentId** (section 2.2.4.4). This element is optional.
- **ContentLocation** (section 2.2.4.3). This element is optional.
- **IsInline** (section 2.2.4.9). This element is optional.

2.2.3.3 Body

The **Body** type is an optional child type of the **ApplicationData** element that specifies details about the body of an e-mail, contact, task, or calendar item in a response message. The **Body** type MUST be included in a response message whenever a content class has changes or new items.

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

There MUST be no limit on the number of **Body** types in a command response. Command requests MAY include the **Body** type.

In a response, the **ApplicationData** element MUST be the parent element of the **Body** type. For more information about the **ApplicationData** element, see [MS-ASCMD] section 2.2.2.22.1.6.

The **Body** type MUST have the following child elements in this order (elements noted as optional MAY be in the response):

- **Type** (section 2.2.4.14). This element is required.
- **EstimatedDataSize** (section 2.2.4.7). This element is optional in the response.
- **Truncated** (section 2.2.4.12). This element is optional in the response.
- **Data** (section 2.2.4.5). This element is required.

2.2.3.4 BodyPreference

The **BodyPreference** type is an optional **container** ([MS-ASDTYPE] section 2.8) complex type that sets preference information related to the type and size of information that is returned from searching, synchronizing, or fetching.

A command response **MUST NOT** include **BodyPreference** type. Command requests **MAY** include the **BodyPreference** type.

In a request, the **Options** element **MUST** be the parent element of the **BodyPreference** type. The **BodyPreference** type **MUST** have the following child elements in this order (elements noted as optional **MAY** be in the response):

- **Type** (section 2.2.4.11). This element is required.
- **TruncationSize** (section 2.2.4.12). This element is optional.
- **AllOrNone** (section 2.2.1). This element is optional.

The contents of the **Options** element specifies preferences for all of the content that the user is interested in searching, synchronizing, or retrieving. These preferences **MUST** be set on a per-request basis and **MUST** override any stored information. Because this information is required to process every request, the information can be persisted on the **server** if network load is a concern.

There **MUST** be one explicit **BodyPreference** type specified for each set of preferences. The **client** **MUST** specify sets of preferences for different types of content by using a separate **BodyPreference** type for each **Type**. If multiple content types have the same preferences, then the client **MUST** send a different complex type for each type to prevent ambiguity in specifying the preferences.

2.2.4 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.

Element	Description
AllOrNone	Specifies whether to search, synchronize, or retrieve all or none of the content based on the TruncationSize element.
ContentId	Contains the unique object ID for an attachment.
ContentLocation	Contains the relative URL for an attachment, which is used to match a reference to an inline attachment in an HTML message to the attachment in the attachment table.

ContentType	The type of data returned by the Data element.
Data	The body of the calendar item, contact, document, e-mail, or task.
DisplayName	The display name of the attachment.
EstimatedDataSize	An informational estimate of the size of the complete body of the calendar item, contact, document, e-mail, or task
FileReference	The location of an item on the server to retrieve.
IsInline	Specifies whether the attachment is embedded in the message.
Method	Identifies the method in which the attachment was attached.
NativeBodyType	The original format type of the item.
Truncated	Specifies whether the body of the item has been truncated according to the BodyPreferences indicated by the client .
TruncationSize	Specifies the size, in bytes, of the content that the client wants to search, synchronize, or fetch.
Type	The format type of the body content of the item.

2.2.4.1 AllOrNone

The **AllOrNone** element is an optional child element of the **BodyPreference** type that specifies whether to search, synchronize, or retrieve all or none of the content based on the **TruncationSize** (section 2.2.4.9) element.

The value of this element **MUST** be a **boolean** value ([MS-ASDTYPE] section 2.3). When the value is set to 1 (TRUE) and the content has not been truncated, all of the content **MUST** be searched, synchronized, or retrieved. When the value is set to 1 (TRUE) and the content has been truncated, the content **MUST NOT** be searched, synchronized, or retrieved. When the value is set to 0 (FALSE), the truncated or non-truncated content **MUST** be searched, synchronized, or retrieved.

A command request **MUST** have a maximum of 1 **AllOrNone** value per **BodyPreference** container element. If the **AllOrNone** value is not included in the request, then the truncated or non-truncated content **MUST** be searched, synchronized, or retrieved as if the value was set to 0 (FALSE). The **AllOrNone** element **MUST NOT** be used in command responses.

This element **MUST** be ignored if the **TruncationSize** element is not included.

By default, the **server** **MUST** return the content truncated to the size requested by **TruncationSize**, but in some cases the **client** may not want to receive a truncated response for a particular content type. In this case, the server **SHOULD** choose the next best format in which it can deliver the content. An example is to enable servers to downgrade an appointment body to plain text if the Rich Text Format (RTF) equivalent is larger than the **TruncationSize** because the client cannot process partial RTF.

2.2.4.2 ContentId

The **ContentId** element is an optional child element of the **Attachment** type that contains the unique object ID for an attachment. This element is provided for informational purposes only and can be ignored by the **client**.

The value of this element **MUST** be a **string** value ([MS-ASDTYPE] section 2.1).

A command response **MUST** have a maximum of one **ContentId** element per **Attachment** type. Command requests **MUST NOT** include the **ContentId** element.

The **ContentId** element **MUST** have no child elements.

2.2.4.3 ContentLocation

The **ContentLocation** element is an optional child element of the **Attachment** type that contains the relative URL for an attachment, and is used to match a reference to an inline attachment in an HTML message to the attachment in the attachment table. The **ContentLocation** element **MUST** be included in the response if **IsInline** is set to TRUE (1).

The value of this element **MUST** be a **string** ([MS-ASDTYPE] section 2.1) value.

A command response **MUST** have a maximum of one **ContentLocation** element per **Attachment** type. Command requests **MUST NOT** include the **ContentLocation** element.

The **ContentLocation** element **MUST** have no child elements.

2.2.4.4 ContentType

The **ContentType** element is an optional **string** ([MS-ASDTYPE] section 2.1) element that specifies the type of data returned by the **Data** (section 2.2.1.5) element.

A command response MUST have a maximum of one **ContentType** element. Command requests MUST NOT include the **ContentType** element.

In a response, the **Properties** element of the **Fetch** command MUST be the parent element of the **ContentType** element. The **ContentType** element MUST have no child elements.

2.2.4.5 Data

The **Data** element is a required child element of the **Body** type that contains the data of the body of the calendar item, contact, document, e-mail, or task.

The value of this element MUST be a **string** ([MS-ASDTYPE] section 2.1)

A command response MUST have a maximum of one **Data** element. Command requests MUST NOT include the **Data** element.

In a response, the **Data** element MUST have no child elements.

The **Body** element SHOULD be returned in a CDATA section in the format that is specified by the **Type** element. If the **Truncated** element is included in the response, then the data in the **Data** element MUST be truncated. The **EstimatedDataSize** value SHOULD provide a rough estimation of the actual size of the complete **Body** string. The **client** SHOULD make appropriate buffer provisions to handle the incoming data. If the **AllOrNone** element of the **Search** command is included, and there is no **BodyType** (also set by the **Search** command) that the **server** can fall back to, then the **Data** element MAY NOT be sent to the client.

2.2.4.6 DisplayName

The **DisplayName** element is an optional child element of the **Body** type that specifies the display name of the attachment.

The value of this element MUST be a **string** value ([MS-ASDTYPE] section 2.1).

A command response MUST have a maximum of one **DisplayName** element. Command requests MUST NOT include the **DisplayName** element.

The **DisplayName** element MUST have no child elements.

2.2.4.7 EstimatedDataSize

The **EstimatedDataSize** element is an optional child element of the **Body** type and a required child element of the **Attachment** type. This element that provides an informational estimate of the size of the complete body of the calendar item, contact, document, e-mail, or task. The **EstimatedDataSize** element SHOULD be included a in a response message whenever the **Truncated** element is set to TRUE.

The value of this element MUST be an **integer** value ([MS-ASDTYPE] section 2.2).

A command response MUST have a maximum of one **EstimatedDataSize** element per **Body** or **Attachment** type. Command requests MUST NOT include the **EstimatedDataSize** element. If an **EstimatedDataSize** element is included in a request, then the element is ignored and no error is thrown.

The **EstimatedDataSize** element MUST have no child elements.

The **EstimatedDataSize** value represents the original size of the content in the store and MUST be specified in bytes. The **client** SHOULD use this number only for an informational display to the user when the **Body** is truncated so that the user can take further action. This number is only an estimate and the actual size of the body when fetched can differ based on the content filtering rules applied. The client SHOULD NOT make any memory allocations based on this number for future requests. The presence of this element in a response indicates that the content in the **Data** element is truncated.

2.2.4.8 FileReference

The **FileReference** element specifies the location of an item on the **server** to retrieve. In a request, the **FileReference** element is an optional child element of the **Fetch** element. In a response, the **FileReference** element is a required child element of the **Attachment** type.

The value of this element MUST be a **string** value ([MS-ASDTYPE] section 2.1).

The **FileReference** element MUST have no child elements.

2.2.4.9 IsInline

The **IsInline** element is an optional child element of the **Attachment** type that specifies whether the attachment is embedded in the message.

The value of this element MUST be a **boolean** value ([MS-ASDTYPE] section 2.3).

A command response MUST have a maximum of one **IsInline** element per **Attachment** type. Command requests MUST NOT include the **IsInline** element.

The **IsInline** element MUST have no child elements.

2.2.4.10 Method

The **Method** element is a required child element of the **Attachment** type that identifies the method in which the attachment was attached.

The value of this element MUST be an **enumeration** value ([MS-ASDTYPE] section 2.9).

A command response MUST have a maximum of one **Method** element per **Attachment** type. Command requests MUST NOT include the **Method** element.

The **Method** element MUST have no child elements.

The following table defines the valid values of the **Method** enumeration.

Value	Meaning	Notes
1	Normal attachment	The attachment is a normal attachment. This is the most common value.
2	Reserved	Do not use.
3	Reserved	Do not use.
4	Reserved	Do not use.
5	Embedded message	Indicates that the attachment is an e-mail message, and that the attachment file has an .eml extension.
6	Attach OLE	Indicates that the attachment is an embedded OLE object, such as an inline image.

2.2.4.11 NativeBodyType

The **NativeBodyType** element is a required child element of the **ApplicationData** element in the **Sync** command ([MS-ASCMD] section 2.2.2.22.1.1) that specifies the original format type of the item.

The value of this element **MUST** be an **enumeration** value ([MS-ASDTYPE] section 2.9).

A command response **MUST** have a maximum of one **NativeBodyType** element per **ApplicationData** element. Command requests **MUST NOT** include the **NativeBodyType** element.

The **NativeBodyType** element **MUST** have no child elements.

The following table defines the valid values of the **NativeBodyType** enumeration.

Enumeration Value	Description
1	Plain Text
2	HTML
3	RTF

The **NativeBodyType** and **Type** elements SHOULD have the same value unless the **server** has modified the format of the body to match the **client's** request. The client can specify a preferred body format by using the **Type** element of a **Search** or **Sync** command request.

2.2.4.12 Truncated

The **Truncated** element is an optional child element of the **Body** type that specifies whether the body of the item has been truncated according to the **BodyPreferences** indicated by the **client**.

The value of this element must be a **boolean** value ([MS-ASDTYPE] section 2.3).

A command response MUST have a maximum of one **Truncated** element per **Body** type. If the command response does not contain a **Truncated** element, then the client MUST process the message as if the whole body is contained in the message response (as if the **Truncated** is set to FALSE). Command requests MUST NOT include the **Truncated** element.

The **Truncated** element MUST have no child elements.

2.2.4.13 TruncationSize

The **TruncationSize** element is an optional child element of the **BodyPreferences** type that specifies the size, in bytes, of the content that the user wants to search, synchronize, or fetch.

The value of this element MUST be an **integer** value ([MS-ASDTYPE] section 2.2).

A command request MUST have a maximum of one **TruncationSize** element per **BodyPreference** type. Command responses MUST NOT include the **TruncationSize** element.

The **TruncationSize** element MUST have no child elements.

If the **TruncationSize** element is absent, then the **server** SHOULD assume that the user will search the entire content. If the server sends the response in the corresponding content type and if the size of the entire content is greater than the value that is specified by **TruncationSize**, then the server MUST truncate the message to the size that is specified by **TruncationSize** and send it in the specified encoding. The maximum value for **TruncationSize** is 2,147,483,647.

2.2.4.14 Type

The **Type** element indicates the format type of the body content of the item. In a request, the **Type** element is a required child element of the **BodyPreferences** type. In a response, the **Type** element is a required child element of the **Body** type.

The value of this element MUST be an **enumeration** value ([MS-ASDTYPE] section 2.9).

A command request or response MUST have a maximum of one **Type** element per **BodyPreference** or **Body** type.

The **Type** element **MUST** have no child elements.

The following table defines the valid values of the **Type** enumeration.

Enumeration Value	Description
1	Plain text
2	HTML
3	RTF
4	MIME

Additional Type values **MAY** be used if the **client** and the **server** support the content type. If the server or client receives a Type value that it does not support or understand, it **SHOULD** skip the entire parent element without processing it.

2.2.5 Attributes

This specification does not define any common **XML** schema attribute definitions.

2.2.6 Groups

The following table summarizes the set of common **XML** schema group definitions defined by this specification. XML schema groups that are specific to a particular operation are described with the operation.

Attribute	Description
TopLevelSchemaProps	Identifies the Body , Attachment , and BodyPreference types as being part of the TopLevelSchemaProps group.

2.2.6.1 TopLevelSchemaProps

The **TopLevelSchemaProps** group identifies the **Body** and **Attachment** types as being part of the **TopLevelSchemaProps** group.

2.2.7 Attribute Groups

This specification does not define any common **XML** schema attribute group definitions.

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 specified in this document.

3.1.2 Timers

None

3.1.3 Initialization

None

3.1.4 Higher-Layer Triggered Events

None

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Commands

Command	Description
Fetch	Retrieves an item from the server.
Search	Retrieves entries from the store.
Sync	Synchronizes changes in a collections set between the client and the server.

3.1.5.1.1 Fetch

The request message for the **Fetch** command MAY include the following elements and types: **BodyPreference**, **Type**, **TruncationSize**, **FileReference**, and **AllOrNone**.

3.1.5.1.2 Search

The request message for the **Search** command MAY include the following elements and types: **BodyPreference**, **Type**, **TruncationSize**, and **AllOrNone**.

3.1.5.1.3 Sync

The request message for the **Sync** command MAY include the following elements and types: **BodyPreference**, **Type**, **TruncationSize**, **AllOrNone**, and **NativeBodyType**.

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 specified in this document.

The abstract data model used by the server and the client are the same.

3.2.2 Timers

None

3.2.3 Initialization

None

3.2.4 Higher-Layer Triggered Events

None

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Commands

Command	Description
----------------	--------------------

Sync	Synchronizes changes in a collections set between the client and the server.
-------------	---

3.2.5.1.1 Sync

The response message for the **Sync** command MAY include the following: Attachment, DisplayName, FileReference, Method, EstimatedDataSize, ContentId, ContentLocation, IsInline, Body, Type, EstimatedDataSize, Truncated, Data, and NativeBodyType.

4 Protocol Examples

See [MS-ASCMD] sections 4.1 and 4.4 for examples of the **Fetch** and **Search** commands utilizing the AirSyncBase Namespace protocol.

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 2003 with Service Pack 3 applied
- Exchange 2003 with Service Pack 2 applied
- 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.

Index

Introduction, 4

Messages, 6

Office/Exchange Behavior, 20

Protocol Details, 18

 Client Details, 18

 Server Details, 19

Protocol Examples, 20

Security, 20