[MS-ASAIRS]:
ActiveSync AirSyncBase Namespace Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0.0	Major	Initial Release.
03/04/2009	1.0.1	Editorial	Revised and edited technical content.
04/10/2009	2.0.0	Major	Updated and revised the technical content.
07/15/2009	3.0.0	Major	Updated and revised the technical content.
11/04/2009	3.1.0	Minor	Updated the technical content.
02/10/2010	3.1.0	None	Version 3.1.0 release

Table of Contents

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]:

attachment
attachment table
calendar
class
collection
contact
display name
embedded message
Hypertext Markup Language (HTML)
Hypertext Transfer Protocol (HTTP)
identifier
message
MIME
Rich Text Format (RTF)
plain text
property
rules
store
Uniform Resource Locator (URL)
XML
XML schema definition (XSD)

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

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

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

[MS-ASDTYPE] Microsoft Corporation, "ActiveSync Data Types", 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.

[XMLNS] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, http://www.w3.org/TR/REC-xml-names/.

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

To use the properties in the AirSyncBase namespace, 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 4.5.1.1.

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 2.2.2.14.

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:element name="Restriction" minOccurs="0" type="xs:string" />
                <xs:element name="Preview" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:unsignedInt">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="255"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </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: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="airsyncbase:EmptyTag"/>
              <xs:element name="Attachments" type="airsyncbase:EmptyTag"/>
          </xs:choice>
      </xs:sequence>
  </xs:group>
</xs:schema>
```

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.

2.2.1 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
Attachments	A collection of elements that contains one or more Attachment items.
Attachment	Specifies the **attachment** information for a single attachment item.
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.

2.2.1.1 Attachments

The Attachments type is an optional child element of the Properties element ([MS-ASCMD] section 2.2.1.8.3.10) in the **Fetch** command that contains one or more attachment items.

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

The Attachments type, if present, MUST have at least one child Attachment type.

2.2.1.2 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 one or more of the following child elements in any order (elements noted as optional can be in the response):

- <DisplayName> (section 2.2.2.5). This element is optional.

- <FileReference> (section 2.2.2.7). This element is required.

- <Method> (section 2.2.2.9). This element is required.

- <EstimatedDataSize> (section 2.2.2.6). This element is required.

- <ContentId> (section 2.2.2.2). This element is optional.

- <ContentLocation> (section 2.2.2.3). This element is optional.

- <IsInline> (section 2.2.2.8). This element is optional.

2.2.1.3 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 is no limit on the number of **Body** types in a command response. Command requests can include the **Body** type.

In a response, the <ApplicationData> element MUST be the parent element of the **Body** type. For more details about the <ApplicationData> element, see [MS-ASCMD] section 2.2.1.19.1.7.

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

- <Type> (section 2.2.2.14). This element is required.

- <EstimatedDataSize> (section 2.2.2.6). This element is optional in the response.

- <Truncated> (section 2.2.2.12). This element is optional in the response.

- <Data> (section 2.2.2.4). This element is required.

2.2.1.4 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 can include the **BodyPreference** type.

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

- <Type> (section 2.2.2.14). This element is required.

- <TruncationSize> (section 2.2.2.13). This element is optional.

- <AllOrNone> (section 2.2.2.1). This element is optional.

- <Preview> (section 2.2.2.15). This element is optional.

The contents of the <Options> element specify preferences for all of the content that the user is interested in searching, synchronizing, or retrieving. These preferences are set on a per-request basis and 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 specifies the 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 sends a different complex type for each <Type> to prevent ambiguity in specifying the preferences.

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.

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**.
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.

Element	Description
Method	Identifies the method in which the attachment was attached.
NativeBodyType	The original format type of the item.
Preview	The length of the message preview to be returned to the client.
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.

2.2.2.1 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 2.2.2.13) element.

The value of this element is 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 is searched, synchronized, or retrieved. When the value is set to 1 (TRUE) and the content has been truncated, the content is not searched, synchronized, or retrieved. When the value is set to 0 (FALSE), the truncated or non-truncated content is 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 is 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 returns the content truncated to the size requested by <TruncationSize>, but in some cases the client does 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.

2.2.2.2 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 is 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.

2.2.2.3 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 value of this element is 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.

2.2.2.4 Data

The <Data> element is a 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 is a **string** ([MS-ASDTYPE] section 2.1)

A command response MUST have a maximum of one <Data> element. Command requests MAY include the <Data> element.

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

The **Body** type is returned as a **string** 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 is truncated. The <EstimatedDataSize> element provides a rough estimation of the actual size of the complete **Body string**. The client makes appropriate buffer provisions to handle the incoming data. If the <AllOrNone> element of the **Search** command is included, and there is no **Body** type (also set by the **Search** command) that the server can fall back to, then the <Data> element is not sent to the client.

2.2.2.5 DisplayName

The <DisplayName> element is an optional child element of the Attachment type that specifies the display name of the attachment.

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

A command response MUST have a maximum of one <DisplayName> element per Attachment type. Command requests MUST NOT include the <DisplayName> element.

The <DisplayName> element MUST have no child elements.

2.2.2.6 EstimatedDataSize

The <EstimatedDataSize> element is an optional child element of the **Body** type and a required child element of the Attachment type. This element 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 in a response message whenever the <Truncated> element is set to TRUE.

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

A command response MUST have a maximum of one <EstimatedDataSize> element per **Body** or Attachment type. 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 is specified in bytes. The client uses 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 does 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.

2.2.2.7 FileReference

The <FileReference> element specifies a unique **identifier** that is assigned by the server to each attachment to a message. In an **ItemOperations** request ([MS-ASCMD] section 2.2.1.8), the <FileReference> element is an optional child element of the <Fetch> element ([MS-ASCMD] section 2.2.1.8.2.2). In a **Sync** command response, ([MS-ASCMD] section 2.2.1.19) the <FileReference> element is a required child element of the **Attachment** type (section 2.2.1.2).

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

The <FileReference> element MUST have no child elements.

2.2.2.8 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 is 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.

2.2.2.9 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 is 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.

Value	Meaning	Notes
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.

2.2.2.10   NativeBodyType

The <NativeBodyType> element is a required child element of the **ApplicationData** element ([MS-ASCMD] section 2.2.1.19.1.7) in the **Sync** command that specifies the original format type of the item.

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

A command response MUST have a maximum of one <NativeBodyType> element per **ApplicationData** element. Command requests MAY 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 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.

2.2.2.11   Preview

The <Preview> element <1> is an optional child element of the **BodyPreference** type. In the request, <Preview> provides the length of the message preview to be returned to the client. In the response, Preview contains the number of characters specified in the request.

The value of this element is an **integer** value ([MS-ASDTYPE] section 2.2). This value can be set from 0 to 255, inclusive.

A command request MUST have a maximum of one <Preview> element per <Preview> type. Command responses MUST have a maximum of one <Preview> element per **Body** type.

The <Preview> element MUST have no child elements.

2.2.2.12   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 is 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.

2.2.2.13   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 is an **integer** value ([MS-ASDTYPE] section 2.2).

A command request MUST have a maximum of one <TruncationSize> element per **BodyPreferences** 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 assumes 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 truncates the message to the size that is specified by <TruncationSize> and sends it in the specified encoding. The maximum value for <TruncationSize> is 4,294,967,295.

2.2.2.14   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 is an **enumeration** value ([MS-ASDTYPE] section 2.9).

A command request or response MUST have a maximum of one <Type> element per **BodyPreferences** 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 can 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 skips the entire parent element without processing it.

2.2.2.15 Preview

The preview concept is communicated from the client to the server with the newly-defined <Preview> tag, which is a child element of the **BodyPreference** type (section 2.2.1.4). It is a child element of the **body** type (section 2.2.1.3) in server responses. This element is used to delineate how many Unicode characters of a plain text preview are needed up to a maximum of 255 characters. If a client device supports different kinds of encodings (RTF and HTML for example) and desires a <Preview> if it gets a **body** (section 2.2.1.3) of any kind, then it SHOULD include the <Preview> tag with each **BodyPreference** type (section 2.2.1.4). <Preview> has no child elements and is an **integer**.

2.2.3 Attributes

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

2.2.4 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.

2.2.4.1 TopLevelSchemaProps

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

2.2.5 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 can 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 can include the following elements and types: **BodyPreference**, <Type>, <TruncationSize>, <FileReference>, and <AllOrNone>.

3.1.5.1.3   Sync

The request Message for the **Sync** command can include the following elements and types: **BodyPreference**, <Type>, <TruncationSize>, <FileReference>, and <AllOrNone>.

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 can include the following: attachment, <DisplayName>, <FileReference>, <Method>, <EstimatedDataSize>, <ContentId>, <ContentLocation>, <IsInline>, **Body**, <Type>, <Truncated>, <Data>, and <NativeBodyType>.

4 Protocol Examples

See [MS-ASCMD] section 4.5.3.1 and [MS-ASCMD] section 4.5.3.2 for examples of the **Search** command utilizing the AirSyncBase Namespace protocol. For examples of the **Fetch** command utilizing the AirSyncBase Namespace protocol, see [MS-ASCMD] section 4.5.1.1 and [MS-ASCMD] section 4.5.2.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft Exchange Server 2007

- Microsoft Exchange Server 2010

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

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

<1> Section 2.2.2.11: The Preview element is not supported by Exchange Server 2007.

7 Change Tracking

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

8 Index