[MS-ASCNTC]:
ActiveSync Contact Class Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0		Initial Release.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.

Table of Contents

1   Introduction

Mobile devices that communicate using the ActiveSync Protocol need to exchange **contact** data. The Contact **class** protocol specifies the format used by the ActiveSync protocol for the interchange of contact data.

1.1   Glossary

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

> **alias**
> **class**
> **collection**
> **contact**
> **distinguished name (DN)**
> **ghosted**
> **.jpg**
> **Multimedia Messaging Service (MMS)**
> **recipient information cache**
> **synchronization**
> **WAP Binary XML (WBXML)**
> **XML**
> **XML schema**

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-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-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.

[RFC822] Crocker, D.H., "Standard for ARPA Internet Text Messages", RFC 822, August 1982, http://www.ietf.org/rfc/rfc0822.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

The Contact class protocol specifies an **XML** representation of contact data as used in various ActiveSync commands.

1.4 Relationship to Other Protocols

The Contact class protocol specifies XML representation of contact data that is used by commands specified in [MS-ASCMD].

Some types and elements in the calendar class support being **ghosted**. The use of ghosted **properties** is specified in [MS-ASCMD] Section 2.2.1.19.1.12.

All data types in this document conform to the data type definitions specified in [MS-ASDTYPE]. Common **XML schema** elements used by other classes are defined in [MS-ASAIRS].

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 Contact class consists of a series of XML elements that are embedded inside of a **collection** transmitted according to the syntax of the ActiveSync protocol command used. The XML block containing the class elements is transmitted in either the request body of a request, or the response body of a response.

The types and elements of the contact class are defined in three namespaces: contacts, Contacts2, and AirSyncBase. All of the contact class types and elements are specified in this document. However, complex types and elements defined in the AirSyncBase namespace are further specified in [MS-ASAIRS].

The parent element of the contact class elements depends upon the ActiveSync protocol command used to retrieve class data. Commands and parent elements for the contact class XML schema are specified in section 3.1.5.

This section describes the child elements that can be returned by an ActiveSync command. This section also describes concepts that are related to the contact class.

2.2 Message Syntax

The markup MUST be well-formed XML, as specified in [XML].

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

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

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="CONTACTS:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="CONTACTS:"
xmlns:xs=http://www.w3.org/2001/XMLSchema xmlns:A="AirSyncBase:">
     <xs:import namespace="AirSyncBase" />
     <xs:element name="Anniversary" type="xs:dateTime" />
     <xs:element name="AssistantName" type="xs:string" />
     <xs:elementname="AssistantPhoneNumber" type="xs:string" />
     <xs:element name="AssistnamePhoneNumber" type="xs:string" />
     <xs:element name="Birthday" type="xs:dateTime" />
     <xs:element name="Business2PhoneNumber" type="xs:string" />
     <xs:element name="BusinessAddressCity" type="xs:string" />
     <xs:element name="BusinessPhoneNumber" type="xs:string" />
     <xs:element name="WebPage" type="xs:string" />
     <xs:element name="BusinessAddressCountry" type="xs:string" />
     <xs:element name="Department" type="xs:string" />
     <xs:element name="Email1Address" type="xs:string" />
     <xs:element name="Email2Address" type="xs:string" />
     <xs:element name="Email3Address" type="xs:string" />
     <xs:element name="BusinessFaxNumber" type="xs:string" />
     <xs:element name="FileAs" type="xs:string" />
     <xs:element name="Alias" type="xs:string" />
     <xs:element name="WeightedRank" type="xs:int" />
     <xs:element name="FirstName" type="xs:string" />
     <xs:element name="MiddleName" type="xs:string" />
     <xs:element name="HomeAddressCity" type="xs:string" />
     <xs:element name="HomeAddressCountry" type="xs:string" />
```

```xml
        <xs:element name="HomeFaxNumber" type="xs:string" />
        <xs:element name="HomePhoneNumber" type="xs:string" />
        <xs:element name="Home2PhoneNumber" type="xs:string" />
        <xs:element name="HomeAddressPostalCode" type="xs:string" />
        <xs:element name="HomeAddressState" type="xs:string" />
        <xs:element name="HomeAddressStreet" type="xs:string" />
        <xs:element name="MobilePhoneNumber" type="xs:string" />
        <xs:element name="Suffix" type="xs:string" />
        <xs:element name="CompanyName" type="xs:string" />
        <xs:element name="OtherAddressCity" type="xs:string" />
        <xs:element name="OtherAddressCountry" type="xs:string" />
        <xs:element name="CarPhoneNumber" type="xs:string" />
        <xs:element name="OtherAddressPostalCode" type="xs:string" />
        <xs:element name="OtherAddressState" type="xs:string" />
        <xs:element name="OtherAddressStreet" type="xs:string" />
        <xs:element name="PagerNumber" type="xs:string" />
        <xs:element name="Title" type="xs:string" />
        <xs:element name="BusinessAddressPostalCode" type="xs:string" />
        <xs:element name="LastName" type="xs:string" />
        <xs:element name="Spouse" type="xs:string" />
        <xs:element name="BusinessAddressState" type="xs:string" />
        <xs:element name="BusinessAddressStreet" type="xs:string" />
        <xs:element name="JobTitle" type="xs:string" />
        <xs:element name="YomiFirstName" type="xs:string" />
        <xs:element name="YomiLastName" type="xs:string" />
        <xs:element name="YomiCompanyName" type="xs:string" />
        <xs:element name="OfficeLocation" type="xs:string" />
        <xs:element name="RadioPhoneNumber" type="xs:string" />
        <xs:element name ="Picture" type="xs:string"/>
        <xs:element name="Categories">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="Category" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Children">
            <xs:complexType>
                <xs:sequence minOccurs="0">
                    <xs:element maxOccurs="300" name="Child" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="Body" type="A:Body" />
    </xs:schema>
```

The XML schema definition for the Contact2 class in ActiveSync is as follows.

```xml
<?xml version="1.0" ?>
<xs:schema xmlns:tns="CONTACTS2:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="CONTACTS2:"
xmlns:xs=http://www.w3.org/2001/XMLSchema xmlns:A="AirSyncBase:">
    <xs:import namespace="AirSyncBase" />
    <xs:element name ="CustomerId" type="xs:string" />
    <xs:element name ="GovernmentId" type="xs:string"/>
    <xs:element name ="IMAddress" type="xs:string"/>
    <xs:element name ="IMAddress2" type="xs:string"/>
```

```
            <xs:element name ="IMAddress3" type="xs:string"/>
            <xs:element name ="ManagerName" type="xs:string"/>
            <xs:element name ="CompanyMainPhone" type="xs:string"/>
            <xs:element name ="AccountName" type="xs:string" />
            <xs:element name ="NickName" type="xs:string" />
            <xs:element name ="MMS" type="xs:string"/>
    </xs:schema>
```

2.2.1 Complex Types

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

Value	Description
Categories	A collection of categories to which the contact belongs.
Children	A collection of the contact's children.
Body	The notes for this contact.

2.2.1.1 Categories

The <Categories> type is an optional **container** ([MS-ASDTYPE] section 2.8) type that specifies a collection of user labels assigned to the contact.

The <Categories> type has the following child element:

▪ <Categories.Category> (section 2.2.2.61): At least one instance of this element is required.

This type can be ghosted. The use of ghosted properties is specified in [MS-ASCMD] section 2.2.1.19.1.12.

2.2.1.2 Children

The <Children> type is an optional **container** ([MS-ASDTYPE] section 2.8) type that specifies a collection of the contact's children.

The <Children> type has the following child element:

▪ <Child> (section 2.2.2.62): This element is optional.

This type can be ghosted.

2.2.1.3 Body

The <Body> type is an optional **container** ([MS-ASDTYPE] section 2.8) type that specifies the notes for this contact. For more details about the <Body> type, see [MS-ASAIRS].

2.2.2 Elements

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

Contact class elements MUST NOT have child elements in either the command request or response.

The following elements are defined in the Contacts namespace.

Value	Description
Anniversary	The wedding anniversary date for the contact.
AssistantName	The name of the contact's assistant.
AssistantPhoneNumber	The phone number of the contact's assistant.
Birthday	The birth date of the contact.
Business2PhoneNumber	The second business telephone number for the contact.
BusinessAddressCity	The business city of the contact.
BusinessPhoneNumber	The business telephone number for the contact.
WebPage	The Web site or personal Web page for the contact.
BusinessAddressCountry	The business country for the contact.
Department	The department name for the contact.
Email1Address	The first e-mail address for the contact.
Email2Address	The second e-mail address for the contact.
Email3Address	The third e-mail address for the contact.
BusinessFaxNumber	The business fax number for the contact.
FileAs	The filing string – and also, typically, the **display name** – for the contact.
Alias	The user's **alias**.
WeightedRank	The rank this entry possesses in the **recipient information cache**.
FirstName	The contact's first name.
MiddleName	The contact's middle name.
HomeAddressCity	The home city for the contact.
HomeAddressCountry	The home country for the contact.
HomeFaxNumber	The home fax number for the contact.
HomePhoneNumber	The home phone number for the contact.
Home2PhoneNumber	The second home phone number for the contact.
HomeAddressPostalCode	The home postal code for the contact.
HomeAddressState	The home state for the contact.
HomeAddressStreet	The home street address for the contact.

Value	Description
MobilePhoneNumber	The mobile phone number for the contact.
Suffix	The suffix for the contact's name.
CompanyName	The company name for the contact.
OtherAddressCity	The city of the contact's alternative address.
OtherAddressCountry	The country of the contact's alternative address.
CarPhoneNumber	The car telephone number for the contact.
OtherAddressPostalCode	The postal code of the contact's alternative address.
OtherAddressState	The state of the contact's alternative address.
OtherAddressStreet	The street address of the contact's alternative address.
PagerNumber	The pager number for the contact.
Title	The contact's business title.
BusinessAddressPostalCode	The business postal code for the contact.
LastName	The contact's last name.
Spouse	The name of the contact's spouse.
BusinessAddressState	The business state for the contact.
BusinessAddressStreet	The business street address for the contact.
JobTitle	The contact's job title.
YomiFirstName	The Japanese phonetic rendering of the first name of the contact.
YomiLastName	The Japanese phonetic rendering of the last name of the contact.
YomiCompanyName	The Japanese phonetic rendering of the company name for the contact.
OfficeLocation	The office location for the contact.
RadioPhoneNumber	The radio telephone number for the contact.
Picture	The Base64-encoded **.jpg** file containing the picture of the contact.
Categories.Category	A category to which the contact is assigned.
Children.Child	One of the child contacts underneath this contact.

The following elements are defined in the Contacts2 namespace.

Value	Description
CustomerId	The customer identifier (ID) for the contact.
GovernmentId	The government-assigned identifier (ID) for the contact.

Value	Description
IMAddress	The instant messaging address for the contact.
IMAddress2	The alternative instant messaging address for the contact.
IMAddress3	The tertiary instant messaging address for the contact.
ManagerName	The **distinguished name (DN)** of the manager for the contact.
CompanyMainPhone	The main telephone number for the contact's company.
AccountName	The account name and/or number for the contact.
NickName	The nickname for the contact.
MMS	The Multimedia Messaging Service (MMS) address for the contact.

2.2.2.1 Anniversary

The <Anniversary> element is an optional element that specifies the wedding anniversary date for the contact. It is defined as an element in the **Contacts** namespace.

This element is ghosted.

2.2.2.2 AssistantName

The <AssistantName> element is an optional element that specifies the name of the contact's assistant. It is defined as an element in the Contacts namespace.

This element is ghosted.

2.2.2.3 AssistantPhoneNumber

The <AssistantPhoneNumber> element is an optional element that specifies the phone number of the contact's assistant. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.4 Birthday

The <Birthday> element is an optional element that specifies the birth date of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.5 Business2PhoneNumber

The <Business2PhoneNumber> element is an optional element that specifies the secondary business telephone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.6 BusinessAddressCity

The <BusinessAddressCity> element is an optional element that specifies the business city of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.7 BusinessPhoneNumber

The <BusinessPhoneNumber> element is an optional element that specifies the primary business phone number for the contact. It is defined as an element in the contacts namespace.

This element can be ghosted.

2.2.2.8 WebPage

The <WebPage> element is an optional element that specifies the Web site or personal Web page for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.9 BusinessAddressCountry

The <BusinessAddressCountry> element is an optional element that specifies the business country of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.10 Department

The <Department> element is an optional element that specifies the department name for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.11 Email1Address

The <Email1Address> element is an optional element that specifies the first e-mail address for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

The <Email1Address> element is one of the Contact elements returned in a recipient information cache request. For more details about tnteraction with the recipient information cache, see [MS-ASCMD] section 2.2.1.4.2.11.

2.2.2.12 Email2Address

The <Email2Address> element is an optional element that specifies the second e-mail address for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.13 Email3Address

The <Email3Address> element is an optional element that specifies the third e-mail address for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.14 BusinessFaxNumber

The <BusinessFaxNumber> element is an optional element that specifies the business fax number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.15 FileAs

The <FileAs> element is an optional element that specifies the filing name for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

The <FileAs> element is one of the Contact elements returned in a recipient information cache request. For more details about the interaction with the recipient information cache, see [MS-ASCMD] section 2.2.1.4.2.11.

2.2.2.16 Alias

The <Alias> element <1> is an optional element that specifies the user's alias. It is defined as an element in the Contacts namespace.

The **Alias** element is only returned in a recipient information cache request. For more details about the interaction with the recipient information cache, see [MS-ASCMD] section 2.2.1.4.2.11.

2.2.2.17 WeightedRank

The <WeightedRank> element <2> is an optional element that specifies the rank of this contact entry in the recipient information cache. It is defined as an element in the Contacts namespace.

Clients can use the <WeightedRank> element to determine which entries in a recipient information cache list should be displayed first in an auto-completion field. Higher values of the <WeightedRank> element identify the most relevant entries.

The <WeightedRank> element is only returned in a recipient information cache request. For more details about the interaction with the recipient information cache, see [MS-ASCMD] section 2.2.1.4.2.11.

2.2.2.18 FirstName

The <FirstName> element is an optional element that specifies the first name of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.19   MiddleName

The <MiddleName> element is an optional element that specifies the middle name of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.20   HomeAddressCity

The <HomeAddressCity> element is an optional element that specifies the home city for the contact. It is defined as an element in the Contacts namespace.

This element is ghosted.

2.2.2.21   HomeAddressCountry

The <HomeAddressCountry> element is an optional element that specifies the home country for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.22   HomeFaxNumber

The <HomeFaxNumber> element is an optional element that specifies the home fax number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.23   HomePhoneNumber

The <HomePhoneNumber> element is an optional element that specifies the home phone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.24   Home2PhoneNumber

The <Home2PhoneNumber> element is an optional element that specifies the alternative home phone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.25   HomeAddressPostalCode

The <HomeAddressPostalCode> element is an optional element that specifies the home postal code for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.26   HomeAddressState

The <HomeAddressState> element is an optional element that specifies the home state for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.27 HomeAddressStreet

The <HomeAddressStreet> element is an optional element that specifies the home street address for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.28 MobilePhoneNumber

The <MobilePhoneNumber> element is an optional element that specifies the mobile phone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.29 Suffix

The <Suffix> element is an optional element that specifies the suffix for the contact's name. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.30 CompanyName

The <CompanyName> element is an optional element that specifies the company name for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.31 OtherAddressCity

The <OtherAddressCity> element is an optional element that specifies the city for the contact's alternative address. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.32 OtherAddressCountry

The **OtherAddressCountry** element is an optional element that specifies the country of the contact's alternative address. It is defined as an element in the contacts namespace.

This element can be ghosted.

2.2.2.33 CarPhoneNumber

The <CarPhoneNumber> element is an optional element that specifies the car telephone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.34 OtherAddressPostalCode

The <OtherAddressPostalCode> element is an optional element that specifies the postal code of the contact's alternative address. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.35 OtherAddressState

The <OtherAddressState> element is an optional element that specifies the state of the contact's alternative address. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.36 OtherAddressStreet

The <OtherAddressStreet> element is an optional element that specifies the street address of the contact's alternative address. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.37 PagerNumber

The <PagerNumber> element is an optional element that specifies the pager number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.38 Title

The <Title> element is a optional element that specifies the contact's business title. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.39 BusinessPostalCode

The <BusinessPostalCode> element is an optional element that specifies the business postal code for the contact. It is defined as an element in the Contacts namespace.

2.2.2.40 LastName

The <LastName> element is an optional element that specifies the contact's last name. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.41 Spouse

The <Spouse> element is an optional element that specifies the name of the contact's spouse. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.42 BusinessAddressState

The <BusinessAddressState> element is an optional element that specifies the business state for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.43 BusinessAddressStreet

The <BusinessAddressStreet> element is an optional element that specifies the business street address for the contact. It is defined as an element in the Contactsnamespace.

This element can be ghosted.

2.2.2.44 JobTitle

The <JobTitle> element is an optional element that specifies the contact's job title. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.45 YomiFirstName

The <YomiFirstName> element is an optional element that specifies the Japanese phonetic rendering of the first name of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.46 YomiLastName

The <YomiLastName> element is an optional element that specifies the Japanese phonetic rendering of the last name of the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.47 YomiCompanyName

The <YomiCompanyName> element is an optional element that specifies the Japanese phonetic rendering of the company name for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.48 OfficeLocation

The <OfficeLocation> element is an optional element that specifies the office location for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.49 RadioPhoneNumber

The <RadioPhoneNumber> element is an optional element that specifies the radio phone number for the contact. It is defined as an element in the Contacts namespace.

This element can be ghosted.

2.2.2.50 CustomerId

The <CustomerId> element is an optional element that specifies the customer identifier (ID) for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.51 GovernmentId

The <GovernmentId> element is an optional element that specifies the government-assigned identifier (ID) for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.52 IMAddress

The <IMAddress> element is an optional element that specifies the instant messaging address for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.53 IMAddress2

The <IMAddress2> element is an optional element that specifies the alternative instant messaging address for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.54 IMAddress3

The <IMAddress3> element is an optional element that specifies the tertiary instant messaging address for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.55 ManagerName

The <ManagerName> element is an optional element that specifies the distinguished name (DN) of the contact's manager. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.56 CompanyMainPhone

The <CompanyMainPhone> element is an optional element that specifies the main telephone number for the contact's company. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.57 AccountName

The <AccountName> element is an optional element that specifies the account name and/or number for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.58 NickName

The <NickName> element is an optional element that specifies the nickname for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.59   MMS

The <MMS> element is an optional element that specifies the [Multimedia Messaging Service (MMS)](#) address for the contact. It is defined as an element in the Contacts2 namespace.

This element can be ghosted.

2.2.2.60   Picture

The <Picture> element is an optional element that specifies the Base64-encoded .jpg file containing the picture of the contact. It is defined as an element in the Contacts namespace.

The size of the value of <Picture> MUST be limited to 36.6 kilobytes (KB).

2.2.2.61   Categories.Category

The <Categories.Category> element is an optional child element of the **Categories** type that specifies a category to which the contact is assigned. It is defined as an element in the Contacts namespace.

A command request or response has a minimum of one <Categories.Category> element per **Categories** type. It can have up to 300 elements per **Categories** type.

2.2.2.62   Children.Child

The <Children.Child> element is an optional child element of the **Children** type that specifies one of the child contacts underneath this contact. It is defined as an element in the Contacts namespace.

A command request or response has zero or more <Children.Child> elements per **Children** type. It can have up to 300 elements per **Children** type.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

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

Contact class: A structured XML text block that specifies a contact and adheres to the XML schema definition specified in section 2.1. It is returned by the server as part of a full XML response to the client commands specified in section 3.1.5. It is processed by the client as part of the full XML body transmitted by the server in response to the client commands specified in section 3.1.5.

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

3.1.2 Timers

No timer **actions** are defined for the contacts or Contacts2 class.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Synchronizing Contact Data Between Client and Server

A client initiates **synchronization** of contact class data with the server by sending a **Sync** command request.

3.1.4.1.1 Truncating the Contact Notes Field

A client can specify that the Notes contained in a Body text be truncated by sending a <Truncated> element to the server in a **Sync** command. The behavior of <Truncated> is specified in [MS-ASAIRS] section 2.2.2.12.

Once a client device has requested that the Notes field be synchronized using truncation, it cannot request the entire Notes field again.

If a **Body** type is not sent from the client to the server, then the server MUST NOT delete the stored Notes for the contact.

Client devices that do not support Notes functionality for contacts MUST NOT send a **Body** type when synchronizing contact information with a server.

3.1.4.2 Searching a Server for Contacts

A client searches for contact data by sending a **Search** command request to the server.

3.1.4.3 Requesting Details for Specific Contacts

Contact data for one or more additional contacts 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.

3.1.4.4 Refreshing The Recipient Information Cache

A minimal set of contact data is retrieved by the client using a command request against **folder** type 19 (the recipient information cache).

This use of **FolderSync** is further specified in [MS-ASCMD] section 2.2.1.4.

3.1.4.5 Omitting Ghosted Properties from a Sync Change Request

At the beginning of a session (i.e., when the sync key is 0), the client uses the <Supported> element of the **Sync** command request to signify which properties are not ghosted. In subsequent **Sync** requests, the client includes only the set of <Supported> elements from the **Sync** request's <Change> element.

For more information on ghosted properties, see [MS-ASCMD] section 2.2.1.19.1.12.

3.1.5 Message Processing Events and Sequencing Rules

The following section describes how various elements of the contacts class are used in the context of specific ActiveSync commands. Command details are specified in [MS-ASCMD].

3.1.5.1 ItemOperations Command Request

A client uses the **ItemOperations** command to retrieve specific contact items from the server using the <Fetch> element. An **ItemOperations** request can contain multiple <Fetch> elements.

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

Contact class complex types and elements are transmitted as children of the **Schema** type ([MS-ASCMD] section 2.2.1.8.2.12).

ItemOperations is specified in [MS-ASCMD] section 2.2.1.8.

3.1.5.2 Search Command Request

A client uses the **Search** command to retrieve contact class items that match the criteria specified by the client.

The complex types and elements for the contact class MUST NOT be included in a **Search** command request.

Search is specified in [MS-ASCMD] section 2.2.1.14.

3.1.5.3 Sync Command Request

A client uses the **Sync** command to synchronize its contact class items for a specified user with the contacts currently stored by the server.

Any of the complex types and elements for the contact class can be included in a **Sync** command request.

Contact class complex types are transmitted as children of the **ApplicationData** type ([MS-ASCMD] section 2.2.1.19.1.7).

Contact class complex types and elements can be transmitted as children of the **Supported** type ([MS-ASCMD] section 2.2.1.19.1.12) in order to support ghosted elements.

Sync is specified in [MS-ASCMD] section 2.2.1.19.

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.

Contact class **:** A structured XML text block that specifies a contact and adheres to the XML schema definition specified in section 2.1. It is returned by the server as part of a full XML response to the client commands specified in section 3.1.5. It is processed by the client as part of the full XML body transmitted by the server in response to the client commands specified in section 3.1.5.

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

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Synchronizing Contact Data Between Client and Server

A client initiates synchronization of contact data with the server by sending a **Sync** command request. The server responds with a **Sync** command response.

3.2.4.2   Searching a Server for Contacts

A client searches for contact class data by sending a **Search** command request to the server. The server responds with a **Search** command response.

3.2.4.3   Requesting Details for Specific Contacts

Contact data for one or more contact items is requested by the client using an **ItemOperations** command request, which is a wrapper for the <Fetch> element. An **ItemOperations** command can contain multiple <Fetch> elements. The server responds with an **ItemOperations** command response.

3.2.4.4   Refreshing The Recipient Information Cache

A minimal set of contact data is retrieved by the client using a **FolderSync** command request against folder type 19 (the recipient information cache). The server responds with a **FolderSync** command response, and includes only the following elements from the contacts class:

- <Email1Address> (Section 2.2.2.11)

- <FileAs> (Section 2.2.2.15)

- <Alias> (Section 2.2.2.16)

- <WeightedRank> (Section 2.2.2.17)

This use of **FolderSync** is further specified in [MS-ASCMD] section 2.2.1.4.

3.2.4.5   Omitting Ghosted Properties from a Sync Change Request

At the beginning of a session (i.e., when the sync key is 0), the client uses the <Supported> element of the **Sync** command request to signify which properties are not ghosted. In subsequent **FolderSync** requests, the client includes only these elements from the **FolderSync** request's <Change> element. ghosted elements are not sent to the server. Instead of deleting these excluded properties, the server preserves their previous value.

For more information on ghosted properties, see [MS-ASCMD] section 2.2.1.19.1.12.

3.2.5   Message Processing Events and Sequencing Rules

The following section describes how various elements of the contacts class are used in the context of specific ActiveSync commands. Command details are specified in [MS-ASCMD].

3.2.5.1   ItemOperations Command Response

A client uses the **ItemOperations** command to retrieve specific contact items from the server using the <Fetch> element. An **ItemOperations** request can contain multiple **ItemOperations** elements.

Any of the complex types for the contact 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 **ItemOperations** element.

Contact class complex types are returned as children of the **Properties** type ([MS-ASCMD] section 2.2.1.8.3.8).

ItemOperations is specified in [MS-ASCMD] section 2.2.1.8.

3.2.5.2   Search Command Response

A client uses the **Search** command to retrieve contact class items that match the criteria specified by the client.

Any of the complex types and elements for the contact class can be included in a **Search** command request.

Contact class complex types are returned as children of the **Properties** type ([MS-ASCMD] section 2.2.1.14.2.2).

Search is specified in [MS-ASCMD] section 2.2.1.14.

3.2.5.3   Sync Command Response

A client uses the **Sync** command to synchronize its contact class items for a specified user with the contacts currently stored by the server.

Any of the complex types for the contact class can be included in a **Sync** command response.

Contact class complex types and elements are returned as children of the **ApplicationData** type ([MS-ASCMD] section 2.2.1.19.2.2).

Sync is specified in [MS-ASCMD] section 2.2.1.19.

3.2.6   Timer Events

None.

3.2.7   Other Events

None.

4 Protocol Examples

The following request/response transaction demonstrates a client requesting contact synchronization with a server, and the server returning a single new contact. In this example, the complex types and elements of the contact class are contained as children of the **Add** type in the server response.

Note In this example, the value of the <Picture> element, which is a Base64-encoded representation of the image, has been truncated.

Request:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>2006814013</SyncKey>
      <CollectionId>2</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
    </Collection>
  </Collections>
</Sync>
```

Response:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="AirSyncBase:" xmlns:B="POOMCONTACTS:">
  <Collections>
    <Collection>
      <SyncKey>243360144</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <A:Body>
              <A:Type>3</A:Type>
              <A:EstimatedDataSize>5500</A:EstimatedDataSize>
              <A:Truncated>1</A:Truncated>
            </A:Body>
            <B:WebPage>http://www.contoso.com/</B:WebPage>
            <B:BusinessCountry>United States of America</B:BusinessCountry>
            <B:Email1Address>"Rajesh M. Patel (someone@example.com)"
&lt;someone@example.com&gt;</B:Email1Address>
            <B:BusinessFaxNumber>(206) 555-7912</B:BusinessFaxNumber>
            <B:FileAs>Patel, Rajesh M.</B:FileAs>
            <B:FirstName>Rajesh</B:FirstName>
            <B:HomePhoneNumber>(206) 555-9201</B:HomePhoneNumber>
            <B:BusinessCity>Redmond</B:BusinessCity>
            <B:MiddleName>M.</B:MiddleName>
            <B:MobilePhoneNumber>(206) 555-3248</B:MobilePhoneNumber>
            <B:CompanyName>Contoso, Ltd.</B:CompanyName>
            <B:BusinessPostalCode>10021</B:BusinessPostalCode>
            <B:LastName>Patel</B:LastName>
            <B:BusinessState>WA</B:BusinessState>
```

```
                <B:BusinessStreet>234 Main St.</B:BusinessStreet>
                <B:BusinessPhoneNumber>(206) 555-9102</B:BusinessPhoneNumber>
                <B:JobTitle>Development Manager</B:JobTitle>
                <B:Picture>/9j/4AAQSkZJRgABAQEAYABgAAD/...</B:Picture>
                <A:NativeBodyType>3</A:NativeBodyType>
            </ApplicationData>
          </Add>
        </Commands>
      </Collection>
    </Collections>
  </Sync>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following product versions:

- Microsoft Exchange Server 2007

- Microsoft Exchange Server 2010

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

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

<1> Section 2.2.2.16: Exchange 2007 does not support the <Alias> element.

<2> Section 2.2.2.17: Exchange 2007 does not support the <WeightedRank> element.

Release: Sunday, July 12, 2009

7 Change Tracking

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

Release: Sunday, July 12, 2009

8 Index

A

Abstract data model
  client
  server
Applicability

C

Capability negotiation
Change tracking
Client
  abstract data model
  higher-layer triggered events
  initialization
  local events
  message processing
  overview
  sequencing rules
  timer events
  timers

D

Data model - abstract
  client
  server

E

Examples - overview

F

Fields - vendor-extensible

G

Glossary

H

Higher-layer triggered events
  client
  server

I

Initialization
  client
  server
Introduction

L

Local events
  client
  server

M

Message processing
  client
  server
Messages
  overview
  syntax
  transport

N

Normative references

O

Overview (synopsis)

P

Preconditions
Prerequisites
Product behavior

R

References
  normative
Relationship to other protocols

S

Security
  overview
Sequencing rules
  client
  server
Server
  abstract data model
  higher-layer triggered events
  initialization
  local events
  message processing
  overview
  sequencing rules
  timer events
  timers
Standards assignments

T

Timer events
  client
  server
Timers
  client
  server
Tracking changes

Triggered events - higher-layer