

More ActiveSync

Dominic Michael Salemno

Introduction

In my previous ActiveSync blog, entitled [An ActiveSync Primer](#), I delved into the basics of the ActiveSync Protocol. This is the second blog in a series intended to thoroughly explain the ActiveSync Protocol.

Communications

In my previous blog, I stated that ActiveSync uses HTTPS for its communications channel. This is not necessarily true, only mostly☺. The truth is that HTTP can be used in place of HTTPS. This is not as secure as the latter. The transport method needed will depend on the Virtual Directory (VDir) configuration on the server.

Each HTTP POST will contain a series of request and response packets. Each command issued by the client will be in the form of a request. The response of course is the return packet from the server containing the results relevant to the preceding request.

Let's look at an example:

This example fetches an e-mail item from the server.

Request

```
POST /Microsoft-Server-ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=device1&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
```

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync:"
xmlns:airsyncbase="AirSyncBase:" xmlns="ItemOperations:">
  <Fetch>
    <Store>Mailbox</Store>
    <airsync:CollectionId>7</airsync:CollectionId>
    <airsync:ServerId>7:1</airsync:ServerId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
        <airsyncbase:TruncationSize>5120</airsyncbase:TruncationSize>
        <airsyncbase:AllOrNone>0</airsyncbase:AllOrNone>
      </airsyncbase:BodyPreference>
```

```
</Options>
</Fetch>
</ItemOperations>
```

Response

HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 409
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 14.0
Date: Tue, 08 May 2007 17:29:52 GMT

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync:" xmlns:email="POOMMAIL:"
xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <airsync:CollectionId>7</airsync:CollectionId>
      <airsync:ServerId>7:1</airsync:ServerId>
      <airsync:Class>Email</airsync:Class>
      <Properties>
        <email:To>"deviceuser" &lt;someone1@example.com&gt;</email:To>
        <email:Cc>"deviceuser3" &lt;someone3@example.com&gt;</email:Cc>
        <email:From>"deviceuser2" &lt;someone2@example.com&gt;
        </email:From>
        <email:Subject>Subject</email:Subject>
        <email:DateReceived>2007-05-08T17:29:07.890Z
        </email:DateReceived>
        <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
        <email:ThreadTopic>Subject</email:ThreadTopic>
        <email:Importance>1</email:Importance>
        <email:Read>0</email:Read>
        <airsyncbase:Body>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:EstimatedDataSize>20
          </airsyncbase:EstimatedDataSize>
          <airsyncbase:Data>Body as plain text</airsyncbase:Data>
        </airsyncbase:Body>
        <email:MessageClass>IPM.Note</email:MessageClass>
```

```
<email:InternetCPID>28591</email:InternetCPID>
<email:Flag />
<email:ContentClass>urn:content-classes:message
</email:ContentClass>
<airsynibase:NativeBodyType>1</airsynibase:NativeBodyType>
</Properties>
</Fetch>
</Response>
</ItemOperations>
```

The request illustrates using the *fetch* command to obtain an e-mail message from the server. Subsequently, the server sends out a *fetch response* containing the requested e-mail message.

This command as well as other commands is explained in much detail in the [\[MS-ASCMD\]](#) document ([ActiveSync Command Reference Protocol Specification](#)). This document describes all of the various ActiveSync commands that can be sent over the wire.

WBXML

The above examples are shown using the XML structures, not WBXML. This is to simplify matters and not in actuality. WBXML is not used for all communications, but most. In the request above, you'll notice that WBXML is specified in the header:

Content-Type: application/vnd.ms-sync.wbxml

So really, this message should be encoded using WBXML, but the XML equivalent is shown for simplicity's sake. This is done across the ActiveSync specification documents found in [Microsoft's Open Specifications](#).

Overview

To reiterate the concept behind ActiveSync, simplistically, ActiveSync ensures the mobile device in question is constantly sending various request packets to the server (usually an Exchange Server) in an effort to maintain the most up-to-date information on the device. Whether this is simply pulling new information from the server, or sending updated information from the device.

All of the protocol documents describing this protocol in detail can be found on [Microsoft's Open Specifications site](#). Underneath the [Exchange Server Protocols section](#), one will find every document describing this protocol to begin with [MS-AS (AS denoting ActiveSync)]. All of the documents in question are as follows:

[\[MS-ASAIRS\].pdf](#)
[\[MS-ASCAL\].pdf](#)
[\[MS-ASCMD\].pdf](#)
[\[MS-ASCNTC\].pdf](#)

[*\[MS-ASCON\].pdf*](#)
[*\[MS-ASDOC\].pdf*](#)
[*\[MS-ASDTYPE\].pdf*](#)
[*\[MS-ASEMAIL\].pdf*](#)
[*\[MS-ASHTTP\].pdf*](#)
[*\[MS-ASMS\].pdf*](#)
[*\[MS-ASNOTE\].pdf*](#)
[*\[MS-ASPROV\].pdf*](#)
[*\[MS-ASTASK\].pdf*](#)
[*\[MS-ASWBXML\].pdf*](#)

These documents can be daunting, so in an effort to simplify matters, these series will assist those wanting to learn the protocol in more depth. The next blog will walk through initialization and synchronization of a simple session.