# [MS-ASCMD]:
# ActiveSync Command Reference 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.

---



*Copyright © 2010 Microsoft Corporation.*

## Revision Summary

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 12/03/2008 | 1.0.0 | Major | Initial Release. |
| 01/15/2009 | 1.01 | Editorial | Revised and edited technical content. |
| 03/04/2009 | 1.02 | Editorial | Revised and edited technical content. |
| 04/10/2009 | 2.0.0 | Major | Updated technical content and applicable product releases. |
| 07/15/2009 | 3.0.0 | Major | Revised and edited for technical content. |
| 11/04/2009 | 4.0.0 | Major | Updated and revised the technical content. |
| 02/10/2010 | 5.0.0 | Major | Updated and revised the technical content. |
| 05/05/2010 | 6.0.0 | Major | Updated and revised the technical content. |
| 08/04/2010 | 7.0 | Major | Significantly changed the technical content. |
| 11/03/2010 | 8.0 | Major | Significantly changed the technical content. |



*Copyright © 2010 Microsoft Corporation.*

# Contents


**1 Introduction ..... 17**
1.1 Glossary ..... 17
1.2 References ..... 19
1.2.1 Normative References ..... 19
1.2.2 Informative References ..... 20
1.3 Overview ..... 20
1.4 Relationship to Other Protocols ..... 21
1.5 Prerequisites/Preconditions ..... 22
1.6 Applicability Statement ..... 22
1.7 Versioning and Capability Negotiation ..... 22
1.8 Vendor-Extensible Fields ..... 22
1.9 Standards Assignments ..... 22

**2 Messages ..... 23**
2.1 Transport ..... 23
2.2 Message Syntax ..... 23
2.2.1 Namespaces ..... 23
2.2.2 Commands ..... 24
2.2.2.1 Autodiscover ..... 24
2.2.2.1.1 Request ..... 25
2.2.2.1.1.1 Autodiscover ..... 25
2.2.2.1.1.1.1 Request ..... 25
2.2.2.1.1.1.1.1 EMailAddress ..... 26
2.2.2.1.1.1.1.2 AcceptableResponseSchema ..... 26
2.2.2.1.2 Response ..... 26
2.2.2.1.2.1 Autodiscover ..... 28
2.2.2.1.2.1.1 Response ..... 28
2.2.2.1.2.1.1.1 Culture ..... 28
2.2.2.1.2.1.1.2 User ..... 28
2.2.2.1.2.1.1.2.1 DisplayName ..... 28
2.2.2.1.2.1.1.2.2 EMailAddress ..... 29
2.2.2.1.2.1.1.3 Action ..... 29
2.2.2.1.2.1.1.3.1 Redirect ..... 29
2.2.2.1.2.1.1.3.2 Settings ..... 29
2.2.2.1.2.1.1.3.2.1 Server ..... 30
2.2.2.1.2.1.1.3.2.1.1 Type ..... 30
2.2.2.1.2.1.1.3.2.1.2 Url ..... 30
2.2.2.1.2.1.1.3.2.1.3 Name ..... 30
2.2.2.1.2.1.1.3.2.1.4 ServerData ..... 31
2.2.2.1.2.1.1.3.3 Error ..... 31
2.2.2.1.2.1.1.3.3.1 Status ..... 31
2.2.2.1.2.1.1.3.3.2 Message ..... 31
2.2.2.1.2.1.1.3.3.3 DebugData ..... 32
2.2.2.1.2.1.1.3.3.4 ErrorCode ..... 32
2.2.2.1.2.1.1.4 Error ..... 32
2.2.2.1.2.1.1.4.1 ErrorCode ..... 32
2.2.2.1.2.1.1.4.2 Message ..... 33
2.2.2.1.2.1.1.4.3 DebugData ..... 33
2.2.2.2 FolderCreate ..... 33
2.2.2.2.1 Request ..... 33





*Copyright © 2010 Microsoft Corporation.*

2.2.2.2.1.1 FolderCreate ..... 34
2.2.2.2.1.1.1 SyncKey ..... 34
2.2.2.2.1.1.2 ParentId ..... 35
2.2.2.2.1.1.3 DisplayName ..... 35
2.2.2.2.1.1.4 Type ..... 35
2.2.2.2.2 Response ..... 35
2.2.2.2.2.1 FolderCreate ..... 36
2.2.2.2.2.1.1 Status ..... 36
2.2.2.2.2.1.2 SyncKey ..... 37
2.2.2.2.2.1.3 ServerId ..... 38
2.2.2.3 FolderDelete ..... 38
2.2.2.3.1 Request ..... 38
2.2.2.3.1.1 FolderDelete ..... 39
2.2.2.3.1.1.1 SyncKey ..... 39
2.2.2.3.1.1.2 ServerId ..... 39
2.2.2.3.2 Response ..... 40
2.2.2.3.2.1 FolderDelete ..... 40
2.2.2.3.2.1.1 Status ..... 40
2.2.2.3.2.1.2 SyncKey ..... 41
2.2.2.4 FolderSync ..... 42
2.2.2.4.1 Request ..... 42
2.2.2.4.1.1 FolderSync ..... 43
2.2.2.4.1.1.1 SyncKey ..... 43
2.2.2.4.2 Response ..... 43
2.2.2.4.2.1 FolderSync ..... 45
2.2.2.4.2.1.1 Status ..... 45
2.2.2.4.2.1.2 SyncKey ..... 46
2.2.2.4.2.1.3 Changes ..... 47
2.2.2.4.2.1.3.1 Count ..... 47
2.2.2.4.2.1.3.2 Update ..... 47
2.2.2.4.2.1.3.2.1 ServerId ..... 47
2.2.2.4.2.1.3.2.2 ParentId ..... 48
2.2.2.4.2.1.3.2.3 DisplayName ..... 48
2.2.2.4.2.1.3.2.4 Type ..... 48
2.2.2.4.2.1.3.3 Delete ..... 49
2.2.2.4.2.1.3.3.1 ServerId ..... 49
2.2.2.4.2.1.3.4 Add ..... 50
2.2.2.4.2.1.3.4.1 ServerId ..... 50
2.2.2.4.2.1.3.4.2 ParentId ..... 50
2.2.2.4.2.1.3.4.3 DisplayName ..... 50
2.2.2.4.2.1.3.4.4 Type ..... 51
2.2.2.5 FolderUpdate ..... 52
2.2.2.5.1 Request ..... 52
2.2.2.5.1.1 FolderUpdate ..... 53
2.2.2.5.1.1.1 SyncKey ..... 53
2.2.2.5.1.1.2 ServerId ..... 53
2.2.2.5.1.1.3 ParentId ..... 54
2.2.2.5.1.1.4 DisplayName ..... 54
2.2.2.5.2 Response ..... 54
2.2.2.5.2.1 FolderUpdate ..... 55
2.2.2.5.2.1.1 Status ..... 55
2.2.2.5.2.1.2 SyncKey ..... 56
2.2.2.6 GetAttachment ..... 56



Copyright © 2010 Microsoft Corporation.

2.2.2.6.1 Request ..... 57
2.2.2.6.2 Response ..... 57
2.2.2.7 GetItemEstimate ..... 57
2.2.2.7.1 Request ..... 57
2.2.2.7.1.1 GetItemEstimate ..... 58
2.2.2.7.1.1.1 Collections ..... 58
2.2.2.7.1.1.1.1 Collection ..... 58
2.2.2.7.1.1.1.1.1 airsync:SyncKey ..... 59
2.2.2.7.1.1.1.1.2 CollectionId ..... 59
2.2.2.7.1.1.1.1.3 airsync:ConversationMode ..... 59
2.2.2.7.1.1.1.1.4 airsync:Options ..... 60
2.2.2.7.1.1.1.1.4.1 airsync:Class ..... 60
2.2.2.7.1.1.1.1.4.2 airsync:FilterType ..... 60
2.2.2.7.1.1.1.1.4.3 airsync:MaxItems ..... 61
2.2.2.7.2 Response ..... 62
2.2.2.7.2.1 GetItemEstimate ..... 63
2.2.2.7.2.1.1 Response ..... 63
2.2.2.7.2.1.1.1 Status ..... 63
2.2.2.7.2.1.1.2 Collection ..... 64
2.2.2.7.2.1.1.2.1 CollectionId ..... 64
2.2.2.7.2.1.1.2.2 Estimate ..... 64
2.2.2.8 ItemOperations ..... 64
2.2.2.8.1 Delivery of Content Requested by Fetch ..... 65
2.2.2.8.2 Request ..... 67
2.2.2.8.2.1 ItemOperations ..... 70
2.2.2.8.2.1.1 EmptyFolderContents ..... 70
2.2.2.8.2.1.1.1 airsync:CollectionId ..... 70
2.2.2.8.2.1.1.2 Options ..... 71
2.2.2.8.2.1.1.2.1 DeleteSubFolders ..... 71
2.2.2.8.2.1.2 Fetch ..... 71
2.2.2.8.2.1.2.1 Store ..... 72
2.2.2.8.2.1.2.2 airsync:ServerId ..... 72
2.2.2.8.2.1.2.3 airsync:CollectionId ..... 73
2.2.2.8.2.1.2.4 documentlibrary:LinkId ..... 73
2.2.2.8.2.1.2.5 search:LongId ..... 73
2.2.2.8.2.1.2.6 airsyncbase:FileReference ..... 73
2.2.2.8.2.1.2.7 Options ..... 73
2.2.2.8.2.1.2.7.1 Schema ..... 75
2.2.2.8.2.1.2.7.2 Range ..... 75
2.2.2.8.2.1.2.7.3 UserName ..... 76
2.2.2.8.2.1.2.7.4 Password ..... 76
2.2.2.8.2.1.2.7.5 airsync:MIMESupport ..... 76
2.2.2.8.2.1.3 Move ..... 77
2.2.2.8.2.1.3.1 ConversationId ..... 77
2.2.2.8.2.1.3.2 DstFldId ..... 78
2.2.2.8.2.1.3.3 Options ..... 78
2.2.2.8.2.1.3.3.1 MoveAlways ..... 78
2.2.2.8.3 Response ..... 78
2.2.2.8.3.1 ItemOperations ..... 79
2.2.2.8.3.1.1 Status ..... 80
2.2.2.8.3.1.2 Response ..... 81
2.2.2.8.3.1.2.1 Move ..... 81
2.2.2.8.3.1.2.1.1 Status ..... 81



Copyright © 2010 Microsoft Corporation.

2.2.2.8.3.1.2.1.2 ConversationId ..... 82
2.2.2.8.3.1.2.2 EmptyFolderContents ..... 82
2.2.2.8.3.1.2.2.1 Status ..... 83
2.2.2.8.3.1.2.2.2 airync:CollectionId ..... 84
2.2.2.8.3.1.2.3 Fetch ..... 84
2.2.2.8.3.1.2.3.1 Status ..... 84
2.2.2.8.3.1.2.3.2 airsync:Collectionid ..... 85
2.2.2.8.3.1.2.3.3 airsync:ServerId ..... 85
2.2.2.8.3.1.2.3.4 airsync:Class ..... 86
2.2.2.8.3.1.2.3.5 documentlibrary:LinkId ..... 86
2.2.2.8.3.1.2.3.6 Properties ..... 86
2.2.2.8.3.1.2.3.6.1 Range ..... 87
2.2.2.8.3.1.2.3.6.2 Total ..... 87
2.2.2.8.3.1.2.3.6.3 Data ..... 87
2.2.2.8.3.1.2.3.6.4 Part ..... 88
2.2.2.8.3.1.2.3.6.5 Version ..... 88
2.2.2.9 MeetingResponse ..... 88
2.2.2.9.1 Request ..... 88
2.2.2.9.1.1 MeetingResponse ..... 89
2.2.2.9.1.1.1 Request ..... 90
2.2.2.9.1.1.1.1 UserResponse ..... 90
2.2.2.9.1.1.1.2 CollectionId ..... 90
2.2.2.9.1.1.1.3 RequestId ..... 91
2.2.2.9.1.1.1.4 InstanceId ..... 91
2.2.2.9.2 Response ..... 91
2.2.2.9.2.1 MeetingResponse ..... 92
2.2.2.9.2.1.1 Result ..... 92
2.2.2.9.2.1.1.1 RequestId ..... 92
2.2.2.9.2.1.1.2 Status ..... 93
2.2.2.9.2.1.1.3 CalendarId ..... 93
2.2.2.10 MoveItems ..... 94
2.2.2.10.1 Request ..... 94
2.2.2.10.1.1 MoveItems ..... 95
2.2.2.10.1.1.1 Move ..... 95
2.2.2.10.1.1.1.1 SrcMsgId ..... 96
2.2.2.10.1.1.1.2 SrcFldId ..... 96
2.2.2.10.1.1.1.3 DstFldId ..... 96
2.2.2.10.2 Response ..... 96
2.2.2.10.2.1 MoveItems ..... 97
2.2.2.10.2.1.1 Response ..... 97
2.2.2.10.2.1.1.1 SrcMsgId ..... 97
2.2.2.10.2.1.1.2 Status ..... 97
2.2.2.10.2.1.1.3 DstMsgId ..... 98
2.2.2.11 Ping ..... 98
2.2.2.11.1 Request ..... 99
2.2.2.11.1.1 Ping ..... 100
2.2.2.11.1.1.1 HeartbeatInterval ..... 100
2.2.2.11.1.1.2 Folders ..... 101
2.2.2.11.1.1.2.1 Folder ..... 101
2.2.2.11.1.1.2.1.1 Id ..... 101
2.2.2.11.1.1.2.1.2 Class ..... 101
2.2.2.11.2 Response ..... 102
2.2.2.11.2.1 Ping ..... 102



Copyright © 2010 Microsoft Corporation.

2.2.2.11.2.1.1 Status ..... 102
2.2.2.11.2.1.2 Folders ..... 104
2.2.2.11.2.1.2.1 Folder ..... 104
2.2.2.11.2.1.3 MaxFolders ..... 104
2.2.2.11.2.1.4 HeartbeatInterval ..... 104
2.2.2.12 Provision ..... 105
2.2.2.13 ResolveRecipients ..... 105
2.2.2.13.1 Request ..... 105
2.2.2.13.1.1 ResolveRecipients ..... 106
2.2.2.13.1.1.1 To ..... 107
2.2.2.13.1.1.2 Options ..... 107
2.2.2.13.1.1.2.1 CertificateRetrieval ..... 107
2.2.2.13.1.1.2.2 MaxCertificates ..... 108
2.2.2.13.1.1.2.3 MaxAmbiguousRecipients ..... 108
2.2.2.13.1.1.2.4 Availability ..... 108
2.2.2.13.1.1.2.4.1 StartTime ..... 109
2.2.2.13.1.1.2.4.2 EndTime ..... 109
2.2.2.13.1.1.2.5 Picture ..... 109
2.2.2.13.1.1.2.5.1 MaxSize ..... 110
2.2.2.13.1.1.2.5.2 MaxPictures ..... 110
2.2.2.13.2 Response ..... 110
2.2.2.13.2.1 ResolveRecipients ..... 112
2.2.2.13.2.1.1 Status ..... 112
2.2.2.13.2.1.2 Response ..... 112
2.2.2.13.2.1.2.1 To ..... 112
2.2.2.13.2.1.2.2 Status ..... 113
2.2.2.13.2.1.2.3 RecipientCount ..... 113
2.2.2.13.2.1.2.4 Recipient ..... 114
2.2.2.13.2.1.2.4.1 Type ..... 114
2.2.2.13.2.1.2.4.2 DisplayName ..... 114
2.2.2.13.2.1.2.4.3 EmailAddress ..... 114
2.2.2.13.2.1.2.4.4 Availability ..... 114
2.2.2.13.2.1.2.4.4.1 Status ..... 115
2.2.2.13.2.1.2.4.4.2 MergedFreeBusy ..... 115
2.2.2.13.2.1.2.4.5 Certificates ..... 116
2.2.2.13.2.1.2.4.5.1 Status ..... 117
2.2.2.13.2.1.2.4.5.2 CertificateCount ..... 117
2.2.2.13.2.1.2.4.5.3 RecipientCount ..... 117
2.2.2.13.2.1.2.4.5.4 Certificate ..... 118
2.2.2.13.2.1.2.4.5.5 MiniCertificate ..... 118
2.2.2.13.2.1.2.4.6 Picture ..... 118
2.2.2.13.2.1.2.4.6.1 Status ..... 118
2.2.2.13.2.1.2.4.6.2 Data ..... 119
2.2.2.14 Search ..... 119
2.2.2.14.1 Request ..... 121
2.2.2.14.1.1 Search ..... 124
2.2.2.14.1.1.1 Store ..... 124
2.2.2.14.1.1.1.1 Name ..... 125
2.2.2.14.1.1.1.2 Query ..... 125
2.2.2.14.1.1.1.2.1 And ..... 126
2.2.2.14.1.1.1.2.1.1 FreeText ..... 126
2.2.2.14.1.1.1.2.1.2 airsync:Class ..... 126
2.2.2.14.1.1.1.2.1.3 airsync:CollectionId ..... 127



Copyright © 2010 Microsoft Corporation.

2.2.2.14.1.1.1.2.1.4 ConversationId ..... 127
2.2.2.14.1.1.1.2.1.5 GreaterThan ..... 127
2.2.2.14.1.1.1.2.1.5.1 email:DateReceived ..... 128
2.2.2.14.1.1.1.2.1.5.2 Value ..... 128
2.2.2.14.1.1.1.2.1.6 LessThan ..... 128
2.2.2.14.1.1.1.2.1.6.1 email:DateReceived ..... 129
2.2.2.14.1.1.1.2.1.6.2 Value ..... 129
2.2.2.14.1.1.1.2.2 EqualTo ..... 129
2.2.2.14.1.1.1.2.2.1 documentlibrary:LinkId ..... 129
2.2.2.14.1.1.1.2.2.2 Value ..... 130
2.2.2.14.1.1.1.3 Options ..... 130
2.2.2.14.1.1.1.3.1 airsync:MIMESupport ..... 131
2.2.2.14.1.1.1.3.2 Range ..... 132
2.2.2.14.1.1.1.3.3 Username ..... 133
2.2.2.14.1.1.1.3.4 Password ..... 133
2.2.2.14.1.1.1.3.5 DeepTraversal ..... 133
2.2.2.14.1.1.1.3.6 RebuildResults ..... 133
2.2.2.14.1.1.1.3.7 Picture ..... 134
2.2.2.14.1.1.1.3.7.1 MaxSize ..... 134
2.2.2.14.1.1.1.3.7.2 MaxPictures ..... 135
2.2.2.14.2 Response ..... 135
2.2.2.14.2.1 Search ..... 141
2.2.2.14.2.1.1 Status ..... 141
2.2.2.14.2.1.2 Response ..... 141
2.2.2.14.2.1.2.1 Store ..... 142
2.2.2.14.2.1.2.1.1 Status ..... 142
2.2.2.14.2.1.2.1.2 Result ..... 143
2.2.2.14.2.1.2.1.2.1 airsync:Class ..... 144
2.2.2.14.2.1.2.1.2.2 LongId ..... 144
2.2.2.14.2.1.2.1.2.3 airsync:CollectionId ..... 145
2.2.2.14.2.1.2.1.2.4 Properties ..... 145
2.2.2.14.2.1.2.1.2.4.1 gal:Picture ..... 145
2.2.2.14.2.1.2.1.2.4.1.1 Status ..... 146
2.2.2.14.2.1.2.1.2.4.1.2 gal:Data ..... 146
2.2.2.14.2.1.2.1.3 Range ..... 146
2.2.2.14.2.1.2.1.4 Total ..... 147
2.2.2.15 SendMail ..... 148
2.2.2.15.1 Request ..... 148
2.2.2.15.1.1 SendMail ..... 149
2.2.2.15.1.1.1 ClientId ..... 149
2.2.2.15.1.1.2 AccountId ..... 149
2.2.2.15.1.1.3 SaveInSentItems ..... 150
2.2.2.15.1.1.4 Mime ..... 150
2.2.2.15.2 Response ..... 150
2.2.2.15.2.1 SendMail ..... 151
2.2.2.15.2.1.1 Status ..... 151
2.2.2.16 Settings ..... 151
2.2.2.16.1 Request ..... 152
2.2.2.16.1.1 Settings ..... 155
2.2.2.16.1.1.1 RightsManagementInformation ..... 155
2.2.2.16.1.1.1.1 Get ..... 155
2.2.2.16.1.1.2 Oof ..... 156
2.2.2.16.1.1.2.1 Get ..... 156



Copyright © 2010 Microsoft Corporation.

2.2.2.16.1.1.2.1.1 BodyType ..... 157
2.2.2.16.1.1.2.2 Set ..... 157
2.2.2.16.1.1.2.2.1 OofState ..... 157
2.2.2.16.1.1.2.2.2 StartTime ..... 158
2.2.2.16.1.1.2.2.3 EndTime ..... 158
2.2.2.16.1.1.2.2.4 OofMessage ..... 158
2.2.2.16.1.1.2.2.4.1 AppliesToInternal ..... 159
2.2.2.16.1.1.2.2.4.2 AppliesToExternalKnown ..... 160
2.2.2.16.1.1.2.2.4.3 AppliesToExternalUnknown ..... 160
2.2.2.16.1.1.2.2.4.4 Enabled ..... 161
2.2.2.16.1.1.2.2.4.5 ReplyMessage ..... 161
2.2.2.16.1.1.2.2.4.6 BodyType ..... 162
2.2.2.16.1.1.3 DevicePassword ..... 162
2.2.2.16.1.1.3.1 Set ..... 162
2.2.2.16.1.1.3.1.1 Password ..... 163
2.2.2.16.1.1.4 DeviceInformation ..... 163
2.2.2.16.1.1.4.1 Set ..... 163
2.2.2.16.1.1.4.1.1 Model ..... 164
2.2.2.16.1.1.4.1.2 IMEI ..... 164
2.2.2.16.1.1.4.1.3 FriendlyName ..... 165
2.2.2.16.1.1.4.1.4 OS ..... 165
2.2.2.16.1.1.4.1.5 OSLanguage ..... 165
2.2.2.16.1.1.4.1.6 PhoneNumber ..... 166
2.2.2.16.1.1.4.1.7 UserAgent ..... 166
2.2.2.16.1.1.4.1.8 EnableOutboundSMS ..... 166
2.2.2.16.1.1.4.1.9 MobileOperator ..... 166
2.2.2.16.1.1.5 UserInformation ..... 167
2.2.2.16.1.1.5.1 Get ..... 167
2.2.2.16.2 Response ..... 167
2.2.2.16.2.1 Settings ..... 169
2.2.2.16.2.1.1 Status ..... 170
2.2.2.16.2.1.2 Oof ..... 170
2.2.2.16.2.1.2.1 Status ..... 171
2.2.2.16.2.1.2.2 Get ..... 171
2.2.2.16.2.1.2.2.1 OofState ..... 172
2.2.2.16.2.1.2.2.2 StartTime ..... 172
2.2.2.16.2.1.2.2.3 EndTime ..... 172
2.2.2.16.2.1.2.2.4 OofMessage ..... 173
2.2.2.16.2.1.2.2.4.1 AppliesToInternal ..... 174
2.2.2.16.2.1.2.2.4.2 AppliesToExternalKnown ..... 174
2.2.2.16.2.1.2.2.4.3 AppliesToExternalUnknown ..... 175
2.2.2.16.2.1.2.2.4.4 Enabled ..... 175
2.2.2.16.2.1.2.2.4.5 ReplyMessage ..... 175
2.2.2.16.2.1.2.2.4.6 BodyType ..... 176
2.2.2.16.2.1.3 DeviceInformation ..... 176
2.2.2.16.2.1.3.1 Status ..... 177
2.2.2.16.2.1.4 DevicePassword ..... 177
2.2.2.16.2.1.4.1 Status ..... 177
2.2.2.16.2.1.5 UserInformation ..... 178
2.2.2.16.2.1.5.1 Status ..... 178
2.2.2.16.2.1.5.2 Get ..... 178
2.2.2.16.2.1.5.2.1 Accounts ..... 179
2.2.2.16.2.1.5.2.1.1 Account ..... 179



Copyright © 2010 Microsoft Corporation.

2.2.2.16.2.1.5.2.1.1.1 AccountId ..... 179
2.2.2.16.2.1.5.2.1.1.2 AccountName ..... 179
2.2.2.16.2.1.5.2.1.1.3 UserDisplayName ..... 179
2.2.2.16.2.1.5.2.1.1.4 SendDisabled ..... 180
2.2.2.16.2.1.5.2.1.1.5 EmailAddresses ..... 180
2.2.2.16.2.1.5.2.1.1.5.1 SMTPAddress ..... 180
2.2.2.16.2.1.5.2.1.1.5.2 PrimarySmtpAddress ..... 180
2.2.2.16.2.1.6 RightsManagementInformation ..... 180
2.2.2.16.2.1.6.1 Status ..... 181
2.2.2.16.2.1.6.2 Get ..... 181
2.2.2.17 SmartForward ..... 181
2.2.2.17.1 Request ..... 182
2.2.2.17.1.1 SmartForward ..... 183
2.2.2.17.1.1.1 ClientId ..... 183
2.2.2.17.1.1.2 Source ..... 184
2.2.2.17.1.1.2.1 FolderId ..... 184
2.2.2.17.1.1.2.2 ItemId ..... 184
2.2.2.17.1.1.2.3 LongId ..... 184
2.2.2.17.1.1.2.4 InstanceId ..... 185
2.2.2.17.1.1.3 AccountId ..... 185
2.2.2.17.1.1.4 SaveInSentItems ..... 185
2.2.2.17.1.1.5 ReplaceMime ..... 185
2.2.2.17.1.1.6 Mime ..... 186
2.2.2.17.2 Response ..... 186
2.2.2.17.2.1 SmartForward ..... 186
2.2.2.17.2.1.1 Status ..... 187
2.2.2.18 SmartReply ..... 187
2.2.2.18.1 Request ..... 188
2.2.2.18.1.1 SmartReply ..... 189
2.2.2.18.1.1.1 ClientId ..... 189
2.2.2.18.1.1.2 Source ..... 190
2.2.2.18.1.1.2.1 FolderId ..... 190
2.2.2.18.1.1.2.2 ItemId ..... 190
2.2.2.18.1.1.2.3 LongId ..... 190
2.2.2.18.1.1.2.4 InstanceId ..... 190
2.2.2.18.1.1.3 AccountId ..... 191
2.2.2.18.1.1.4 SaveInSentItems ..... 191
2.2.2.18.1.1.5 ReplaceMime ..... 191
2.2.2.18.1.1.6 Mime ..... 192
2.2.2.18.2 Response ..... 192
2.2.2.18.2.1 SmartReply ..... 192
2.2.2.18.2.1.1 Status ..... 193
2.2.2.19 Sync ..... 193
2.2.2.19.1 Request ..... 194
2.2.2.19.1.1 Empty Sync Request ..... 202
2.2.2.19.1.2 Sync ..... 202
2.2.2.19.1.2.1 Collections ..... 203
2.2.2.19.1.2.1.1 Collection ..... 203
2.2.2.19.1.2.1.1.1 SyncKey ..... 204
2.2.2.19.1.2.1.1.2 CollectionId ..... 205
2.2.2.19.1.2.1.1.3 Supported ..... 205
2.2.2.19.1.2.1.1.4 DeletesAsMoves ..... 206
2.2.2.19.1.2.1.1.5 GetChanges ..... 207



Copyright © 2010 Microsoft Corporation.

2.2.2.19.1.2.1.1.6 WindowSize ..... 207
2.2.2.19.1.2.1.1.7 ConversationMode ..... 208
2.2.2.19.1.2.1.1.8 Options ..... 209
2.2.2.19.1.2.1.1.8.1 FilterType ..... 210
2.2.2.19.1.2.1.1.8.2 Class ..... 211
2.2.2.19.1.2.1.1.8.3 Conflict ..... 212
2.2.2.19.1.2.1.1.8.4 MIMESupport ..... 213
2.2.2.19.1.2.1.1.8.5 MIMETruncation ..... 213
2.2.2.19.1.2.1.1.8.6 MaxItems ..... 214
2.2.2.19.1.2.1.1.9 Commands ..... 214
2.2.2.19.1.2.1.1.9.1 Change ..... 215
2.2.2.19.1.2.1.1.9.1.1 ServerId ..... 216
2.2.2.19.1.2.1.1.9.1.2 ApplicationData ..... 216
2.2.2.19.1.2.1.1.9.2 Delete ..... 216
2.2.2.19.1.2.1.1.9.2.1 ServerId ..... 217
2.2.2.19.1.2.1.1.9.3 Add ..... 217
2.2.2.19.1.2.1.1.9.3.1 Class ..... 218
2.2.2.19.1.2.1.1.9.3.2 ClientId ..... 218
2.2.2.19.1.2.1.1.9.3.3 ApplicationData ..... 218
2.2.2.19.1.2.1.1.9.4 Fetch ..... 219
2.2.2.19.1.2.1.1.9.4.1 ServerId ..... 219
2.2.2.19.1.2.2 Wait ..... 219
2.2.2.19.1.2.3 HeartbeatInterval ..... 220
2.2.2.19.1.2.4 Partial ..... 221
2.2.2.19.1.2.5 WindowSize ..... 222
2.2.2.19.2 Response ..... 222
2.2.2.19.2.1 Sync ..... 226
2.2.2.19.2.1.1 Status ..... 226
2.2.2.19.2.1.2 Limit ..... 228
2.2.2.19.2.1.3 Collections ..... 228
2.2.2.19.2.1.3.1 Collection ..... 229
2.2.2.19.2.1.3.1.1 SyncKey ..... 230
2.2.2.19.2.1.3.1.2 CollectionId ..... 230
2.2.2.19.2.1.3.1.3 Status ..... 230
2.2.2.19.2.1.3.1.4 Commands ..... 231
2.2.2.19.2.1.3.1.4.1 Delete ..... 231
2.2.2.19.2.1.3.1.4.1.1 Class ..... 231
2.2.2.19.2.1.3.1.4.1.2 ServerId ..... 232
2.2.2.19.2.1.3.1.4.2 SoftDelete ..... 232
2.2.2.19.2.1.3.1.4.2.1 ServerId ..... 232
2.2.2.19.2.1.3.1.4.3 Change ..... 233
2.2.2.19.2.1.3.1.4.3.1 Class ..... 233
2.2.2.19.2.1.3.1.4.3.2 ServerId ..... 233
2.2.2.19.2.1.3.1.4.3.3 ApplicationData ..... 234
2.2.2.19.2.1.3.1.4.4 Add ..... 234
2.2.2.19.2.1.3.1.4.4.1 ServerId ..... 234
2.2.2.19.2.1.3.1.4.4.2 ApplicationData ..... 235
2.2.2.19.2.1.3.1.4.5 Responses ..... 235
2.2.2.19.2.1.3.1.4.5.1 Change ..... 236
2.2.2.19.2.1.3.1.4.5.1.1 Class ..... 236
2.2.2.19.2.1.3.1.4.5.1.2 ServerId ..... 237
2.2.2.19.2.1.3.1.4.5.1.3 Status ..... 237
2.2.2.19.2.1.3.1.4.5.2 Add ..... 237



Copyright © 2010 Microsoft Corporation.

2.2.2.19.2.1.3.1.4.5.2.1 Class ..... 237
2.2.2.19.2.1.3.1.4.5.2.2 ClientId ..... 238
2.2.2.19.2.1.3.1.4.5.2.3 ServerId ..... 238
2.2.2.19.2.1.3.1.4.5.2.4 Status ..... 239
2.2.2.19.2.1.3.1.4.5.3 Fetch ..... 239
2.2.2.19.2.1.3.1.4.5.3.1 ServerId ..... 239
2.2.2.19.2.1.3.1.4.5.3.2 Status ..... 239
2.2.2.19.2.1.3.1.4.5.3.3 ApplicationData ..... 240
2.2.2.19.2.1.3.1.5 MoreAvailable ..... 240
2.2.2.19.2.2 Empty Sync Response ..... 240
2.2.2.20 ValidateCert ..... 241
2.2.2.20.1 Request ..... 241
2.2.2.20.1.1 ValidateCert ..... 242
2.2.2.20.1.1.1 CertificateChain ..... 242
2.2.2.20.1.1.1.1 Certificate ..... 242
2.2.2.20.1.1.2 Certificates ..... 242
2.2.2.20.1.1.2.1 Certificate ..... 243
2.2.2.20.1.1.3 CheckCRL ..... 243
2.2.2.20.2 Response ..... 243
2.2.2.20.2.1 ValidateCert ..... 244
2.2.2.20.2.1.1 Status ..... 244
2.2.2.20.2.1.2 Certificate ..... 245
2.2.2.20.2.1.2.1 Status ..... 246
2.2.3 Common Status Codes ..... 247

**3 Protocol Details ..... 254**
3.1 Common Details ..... 254
3.1.1 Abstract Data Model ..... 254
3.1.2 Timers ..... 254
3.1.3 Initialization ..... 254
3.1.4 Higher-Layer Triggered Events ..... 254
3.1.5 Message Processing Events and Sequencing Rules ..... 254
3.1.5.1 Downloading Policy Settings ..... 254
3.1.5.2 Setting Device Information ..... 255
3.1.5.3 Synchronizing a Folder Hierarchy ..... 256
3.1.5.4 Synchronizing Inbox, Calendar, Contacts, and Tasks Folders ..... 256
3.1.5.5 Receiving and Accepting Meeting Requests ..... 258
3.1.5.6 Handling Status Errors ..... 259
3.1.6 Timer Events ..... 260
3.1.7 Other Local Events ..... 260

**4 Protocol Examples ..... 261**
4.1 Downloading the Current Server Security Policy by Using the Provision Command ..... 261
4.2 Discovering Account Settings by Using the Autodiscover Command ..... 261
4.2.1 Request ..... 262
4.2.2 Response - Case Error ..... 262
4.2.3 Response - Case Redirect ..... 262
4.2.4 Response - Case Server Settings ..... 263
4.2.5 Response - Case Framework Error ..... 264
4.2.6 Response – Case Framework Default ..... 264
4.3 Setting Device Information by Using the Settings Command ..... 265
4.3.1 Request ..... 265
4.3.2 Response ..... 265



Copyright © 2010 Microsoft Corporation.

4.4 Synchronizing Folders by Using the FolderSync Command ..... 265
4.4.1 Request ..... 265
4.4.2 Response ..... 266
4.5 Synchronizing Data by Using the Sync Command ..... 267
4.5.1 Downloading Current Information from the Server ..... 267
4.5.1.1 Request ..... 268
4.5.1.2 Response ..... 268
4.5.2 Fetching an E-Mail by Using the ServerId ..... 268
4.5.2.1 Request ..... 268
4.5.2.2 Response ..... 269
4.5.3 Uploading New ApplicationData to the Server ..... 269
4.5.3.1 Request ..... 269
4.5.3.2 Response ..... 269
4.5.4 Updating ApplicationData on the Server ..... 270
4.5.4.1 Request ..... 270
4.5.4.2 Response ..... 270
4.5.5 Deleting an Item from the Server ..... 271
4.5.5.1 Request ..... 271
4.5.5.2 Response ..... 271
4.5.6 Identifying Acceptance of Partial Collections ..... 271
4.5.7 Identifying Acceptance of MIME Content ..... 272
4.5.7.1 Sync Request With Support for MIME Content ..... 272
4.5.7.2 Sync Response with MIME Content ..... 272
4.5.7.3 Sync Request with BodyPreference and MIME Support ..... 273
4.5.7.4 Sync Response with MIME Support ..... 273
4.5.8 Identifying That More Content is Ready for Download ..... 274
4.5.9 Synchronizing the Calendar Folder ..... 275
4.5.9.1 Initial Request ..... 275
4.5.9.2 Initial Response ..... 275
4.5.9.3 Second Request ..... 276
4.5.9.4 Second Response ..... 276
4.5.10 Empty Sync Request and Response ..... 278
4.6 Sending E-Mail Messages by Using the SendMail Command ..... 279
4.6.1 Request ..... 279
4.6.2 Response ..... 279
4.7 Replying to E-Mail Messages by Using the SmartReply Command ..... 280
4.7.1 Request ..... 280
4.7.2 Response ..... 280
4.8 Pinging the Server for Updates by Using the Ping Command ..... 280
4.8.1 Ping Command Request ..... 280
4.8.2 Ping Command Response ..... 281
4.8.2.1 Typical Response ..... 281
4.8.2.2 Response – Changes Found ..... 281
4.8.2.3 Response – HeartbeatInterval Error ..... 281
4.8.2.4 Response – Folder Error ..... 282
4.9 Retrieving Item Estimates by Using the GetItemEstimate Command ..... 282
4.9.1 Request ..... 282
4.9.2 Response ..... 282
4.10 Fetching E-Mail and Attachments by Using the ItemOperations Command ..... 283
4.10.1 Fetching an E-Mail Item ..... 283
4.10.1.1 Request ..... 283
4.10.1.2 Response ..... 284
4.10.2 Fetching a MIME E-Mail Item ..... 284



Copyright © 2010 Microsoft Corporation.

4.10.2.1 Request ..... 284
4.10.2.2 Response ..... 285
4.10.3 Fetching an E-Mail Item with a LongId ..... 286
4.10.3.1 Search Request ..... 286
4.10.3.2 Search Response ..... 287
4.10.3.3 Fetch Request ..... 288
4.10.3.4 Fetch Response ..... 288
4.10.4 Fetching an Attachment ..... 289
4.10.4.1 Sync Request ..... 289
4.10.4.2 Sync Response ..... 289
4.10.4.3 ItemOperation Request ..... 290
4.10.4.4 ItemOperation Response ..... 291
4.11 Searching for an Item in the Mailbox by Using the Search Command ..... 291
4.11.1 Keyword Search ..... 291
4.11.1.1 Request ..... 291
4.11.1.2 Response ..... 292
4.11.2 Forward a Search Result ..... 293
4.11.3 Keyword Search with MIMESupport ..... 293
4.12 Searching the Global Address List by Using the Search Command ..... 295
4.12.1 Request ..... 295
4.12.2 Response ..... 295
4.13 Working with Folders ..... 296
4.13.1 Creating Folders by Using the FolderCreate Command ..... 296
4.13.1.1 Request ..... 296
4.13.1.2 Response ..... 297
4.13.2 Deleting Folders by Using the FolderDelete Command ..... 297
4.13.2.1 Request ..... 297
4.13.2.2 Response ..... 297
4.13.3 Updating Folders by Using the FolderUpdate Command ..... 298
4.13.3.1 Request ..... 298
4.13.3.2 Response ..... 298
4.13.4 Emptying Folder Contents by Using the ItemOperations Command ..... 298
4.13.4.1 Request ..... 298
4.13.4.2 Response ..... 299
4.14 Moving Items to Another Folder by Using the MoveItems Command ..... 299
4.14.1 Request ..... 299
4.14.2 Response ..... 299
4.15 Creating Meetings ..... 300
4.15.1 Uploading a Meeting to the Server ..... 300
4.15.1.1 Request ..... 300
4.15.1.2 Response ..... 301
4.15.2 Sending Meeting Requests ..... 301
4.15.2.1 Request ..... 302
4.15.2.2 Response ..... 304
4.15.3 Adding a Meeting Request to the Attendee's Inbox Folder ..... 304
4.15.3.1 Request ..... 304
4.15.3.2 Response ..... 304
4.15.4 Adding a Meeting to the Attendee's Calendar Folder ..... 306
4.15.4.1 Request ..... 306
4.15.4.2 Response ..... 306
4.16 Responding to Meeting Requests by Using the MeetingResponse Command ..... 307
4.16.1 Request ..... 307
4.16.2 Response ..... 308



Copyright © 2010 Microsoft Corporation.

4.17 Resolving Recipients and Retrieving Free/Busy Data by Using the ResolveRecipients Command ..... 308
4.17.1 Request ..... 308
4.17.2 Response for a GAL Entry ..... 309
4.17.3 Response for a Contact Entry ..... 309
4.17.4 Retrieving Free/Busy Data By Using the ResolveRecipients Command ..... 309
4.17.4.1 Request to Retrieve Free/Busy Data ..... 310
4.17.4.2 Response with MergedFreeBusy Data ..... 310
4.18 Retrieving and Changing OOF Settings by Using the Settings Command ..... 312
4.18.1 Retrieving OOF Settings ..... 312
4.18.1.1 Request ..... 312
4.18.1.2 Response ..... 312
4.18.2 Turning On the OOF Message ..... 313
4.18.2.1 Request ..... 313
4.18.2.2 Response ..... 313
4.18.3 Turning Off the OOF Message ..... 314
4.18.3.1 Request ..... 314
4.18.3.2 Response ..... 314
4.19 Validating Certificates by Using the ValidateCert Command ..... 314
4.19.1 Request ..... 314
4.19.2 Response ..... 315
4.20 Retrieving User Information by Using the Settings Command ..... 315
4.20.1 Request ..... 315
4.20.2 Response ..... 315
4.21 Setting a Device Password by Using the Settings Command ..... 316
4.21.1 Request ..... 316
4.21.2 Response ..... 316
4.22 Accessing Documents on File Shares and URIs by Using the Search and ItemOperations Commands ..... 316
4.22.1 Issuing a Search for Item Metadata ..... 317
4.22.1.1 Request ..... 317
4.22.1.2 Response ..... 318
4.22.2 Fetching an Item Based on Metadata ..... 319
4.22.2.1 Request ..... 319
4.22.2.2 Response ..... 319
4.23 Using the Supported Element and Ghosted Elements in the Sync Command ..... 320
4.23.1 Initial Folder Sync ..... 320
4.23.1.1 Request ..... 320
4.23.1.2 Response ..... 320
4.23.2 Sync Command ..... 322
4.23.2.1 Request ..... 322
4.23.2.2 Response ..... 322
4.23.3 Sync Contacts ..... 322
4.23.3.1 Request ..... 322
4.23.3.2 Response ..... 323
4.23.4 Sync Client Changes ..... 324
4.23.4.1 Request ..... 324
4.23.4.2 Response ..... 324
4.23.5 Sync Server Changes ..... 324
4.23.5.1 Request ..... 324
4.23.5.2 Response ..... 325
4.24 Moving a Conversation by Using the ItemOperations Command ..... 326
4.24.1 Request ..... 326



Copyright © 2010 Microsoft Corporation.

4.24.2 Response....326

**5 Security....327**
5.1 Security Considerations for Implementers....327
5.2 Index of Security Parameters....327

**6 Appendix A: Product Behavior....328**

**7 Change Tracking....335**

**8 Index....339**



*Copyright © 2010 Microsoft Corporation.*

# 1 Introduction

This document specifies the ActiveSync protocol commands which are used by a client, typically a mobile device, to synchronize and exchange objects with a server. These objects include e-mail **messages**, **Short Message Service (SMS)** messages, **attachments**, **folders**, **contact** information, **meetings**, calendar data, tasks, notes and documents.

## 1.1 Glossary

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

**Active Directory**
**address book**
**address list**
**alias**
**ambiguous name resolution (ANR)**
**appointment**
**ASCII**
**attachment**
**Autodiscover server**
**base64 encoding**
**binary large object (BLOB)**
**body part**
**Calendar folder**
**Calendar object**
**character set**
**class**
**collection**
**contact**
**conversation**
**Deleted Items folder**
**distribution list**
**domain**
**Domain Name System (DNS)**
**Drafts folder**
**folder**
**folder ID (FID)**
**fully qualified domain name (FQDN)**
**ghosted**
**Global Address List (GAL)**
**GUID**
**Hypertext Transfer Protocol (HTTP)**
**Inbox folder**
**journal**
**locale**
**mailbox**
**meeting**
**message**
**message body**
**message database (MDB)**
**message ID (MID)**
**MIME**
**named property**
**organizer**



Copyright © 2010 Microsoft Corporation.

**Out of Office (OOF)**
**Outbox folder**
**Personal Information Manager (PIM)**
**plain text**
**property (1)**
**recipient (2)**
**recipient information cache**
**reminder**
**Root folder**
**S/MIME**
**search folder**
**Secure Sockets Layer (SSL)**
**Sent Items folder**
**Short Message Service (SMS)**
**Simple Mail Transfer Protocol (SMTP)**
**special folder**
**store**
**stream (1)**
**synchronization**
**Transport Neutral Encapsulation Format (TNEF)**
**Uniform Resource Identifier (URI)**
**Uniform Resource Locator (URL)**
**Wireless Application Protocol (WAP) Binary XML (WBXML)**
**XML**
**XML namespace**
**XML schema definition (XSD)**

The following terms are specific to this document:

**certificate authority (CA):** A third party that issues public key certificates. Certificates serve to bind public keys to a user identity. Each user and **certificate authority** may decide whether to trust another user or **CA** for a specific purpose, and whether this trust should be transitive.

**certificate revocation lists (CRL):** A list of certificates that have been revoked by the **certificate authority (CA)** (or certification authority) that issued them (that have not yet expired of their own accord). The list must be cryptographically signed by the **CA** that issues it. Typically, the certificates are identified by serial number. In addition to the serial number for the revoked certificates, the **CRL** also contains the revocation reason for each certificate and the time the certificate was revoked. As described in [RFC3280], two types of **CRLs** commonly exist in the industry. Base **CRLs** keep a complete list of revoked certificates, while delta **CRLs** maintain only those certificates that have been revoked since the last issuance of a base **CRL**. For more information, see section 7.3 of [X509], [MSFT-CRL], and section 5 of [RFC3280].

**Universal Naming Convention (UNC):** A standard naming format for specifying the location of network resources such as shared files or devices on a network. The format is "\\<servername>\<share>\<filename>", where <servername> is a NetBIOS name, **fully qualified domain name (FQDN)**, or IPv4 address; <share> is a logical share point for accessing <servername>; and <filename> is the name of the file or device.

**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.



Copyright © 2010 Microsoft Corporation.

## 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-ASCAL] Microsoft Corporation, "ActiveSync Calendar Class Protocol Specification", December 2008.

[MS-ASCNTC] Microsoft Corporation, "ActiveSync Contact Class Protocol Specification", December 2008.

[MS-ASCON] Microsoft Corporation, "ActiveSync Conversations Protocol Specification", April 2009.

[MS-ASDOC] Microsoft Corporation, "ActiveSync Document Class 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-ASHTTP] Microsoft Corporation, "ActiveSync HTTP Protocol Specification", December 2008.

[MS-ASMS] Microsoft Corporation, "ActiveSync Short Message Service Protocol Specification", April 2009.

[MS-ASNOTE] Microsoft Corporation, "ActiveSync Notes Class Protocol Specification", April 2009.

[MS-ASPROV] Microsoft Corporation, "ActiveSync Provisioning Protocol Specification", December 2008.

[MS-ASRM] Microsoft Corporation, "ActiveSync Rights Management Protocol Specification", August 2010.

[MS-ASTASK] Microsoft Corporation, "ActiveSync Tasks Class Protocol Specification", December 2008.

[MS-ASWBXML] Microsoft Corporation, "ActiveSync WAP Binary XML (WBXML) Protocol Specification", December 2008.

[MS-OXCICAL] Microsoft Corporation, "iCalendar to Appointment Object Conversion Protocol Specification", April 2008.

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol Specification", April 2008.

[MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Data Structure", April 2008.



Copyright © 2010 Microsoft Corporation.

[MSFT-CRL] Komar, B., Kinder, C., Ben-Menahem, A., "Certificate Revocation and Status Checking", January 2006, http://technet.microsoft.com/en-us/library/bb457027.aspx

[MSFT-DNS-SRV] Microsoft Corporation, "A new feature is available that enables Outlook 2007 to use DNS Service Location (SRV) records to locate the Exchange Autodiscover service", August 2007, http://support.microsoft.com/?kbid=940881

[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

[RFC2447] Dawson, F., Mansour, S., and Silverberg, S., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 2447, November 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2447.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[RFC5751] Ramsdell, B., and Turner, S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010, http://www.rfc-editor.org/rfc/rfc5751.txt

[X509] International Telecommunication Union, "Information technology - Open Systems Interconnection - The Directory: Public-Key and attribute certificate frameworks", ITU-T Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

[XMLNS] Bray, T., Hollander, D., Layman, A., Eds., et al., "Namespaces in XML 1.0 (Third Edition)", December 2009, http://www.w3.org/TR/REC-xml-names/

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

### 1.2.2 Informative References

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol Specification", April 2008.

[MSFT-AUTODISCOVER] Masterson, J., Turick, J., Smith IV, R., "White Paper: Exchange 2007 Autodiscover Service", November 2007, http://technet.microsoft.com/en-us/library/bb332063(EXCHG.80).aspx

[MSDN-ADDP] Microsoft Corporation, "Establishing an ActiveSync Desktop-Device Partnership", August 2008, http://msdn.microsoft.com/en-us/library/aa917378.aspx

## 1.3 Overview

This protocol consists of a set of **XML**-based commands that are used by a client device to synchronize and exchange its e-mail, files, and data with a server.

The client first uses the **Autodiscover** command to get a user's account configuration. The client can then view and modify server data related to that account, including e-mail messages and attachments, folders, contacts, and calendar requests.



Copyright © 2010 Microsoft Corporation.

The client then uses the **Provision** command to get and subsequently acknowledge security policy settings from the server.

The next command sent by the client is **FolderSync** to retrieve the folder hierarchy of the user.

This is typically followed by **GetItemEstimate** in order to retrieve the number of changes that need to be downloaded to the client via the first **Sync** request. This is immediately followed by **Sync**, to get a **synchronization** key and then messages from the server. Optionally, **Ping** or hanging **Sync** can then be issued to keep the device up-to-date on any server changes.

The client processes outgoing e-mail using the **SendMail**, **SmartReply**, and **SmartForward** commands. For incoming messages, the client can call the **ItemOperations** command to fetch the message, and then use the **MoveItems** command. **S/MIME**messages are processed with the **ResolveRecipients** and **ValidateCert** commands.

The client calls the **FolderSync**, **FolderCreate**, **FolderUpdate**, and **FolderDelete** commands to update, create, and delete **mailbox**folders on the server.

For meeting requests, the client calls the **MeetingResponse** command.

The client can set and request server parameters with the **Settings** command.

The client uses the **Search** command to find particular items on the server.

## 1.4 Relationship to Other Protocols

The ActiveSync commands specified in this document are sent and received over a **Hypertext Transfer Protocol (HTTP)** connection, as specified in [RFC2616] in an HTTP **POST** method. The information contained in the HTTP **POST** header is specified in [MS-ASHTTP]. The information contained in the HTTP message is sent and received in **Wireless Application Protocol (WAP) Binary XML (WBXML)** format, as specified in [MS-ASWBXML] where the content of the WBXML adheres to the commands specified in this document.

Some of the ActiveSync commands specified in this document are used to synchronize or retrieve more than one **class** of content. For example, the **Ping** command can be used to monitor changes to the e-mail, note, contact, calendar, or task classes. The elements included in the **Ping** command change depending on which content class is being monitored. Because each of the content classes are used by multiple commands, they are specified in individual documents for each content type. The content class specifications are [MS-ASEMAIL], [MS-ASCNTC], [MS-ASDOC], [MS-ASCAL], [MS-ASNOTE], and [MS-ASTASK].

Another document containing elements used by multiple commands is [MS-ASAIRS]. [MS-ASAIRS] specifies the AirSyncBase namespace, which is used by multiple commands to specify the formatting preference of body content, truncation sizes, and other commonly used elements.

This document specifies all of the ActiveSync commands except for the **Provision** command, which is independently specified in [MS-ASPROV].

The **Autodiscover** command is specified in this document, but more details about Autodiscover publishing and lookup are available in [MS-OXDSCLI].

All simple data types in this document conform to the data type definitions specified in [MS-ASDTYPE].

For details about how to control the view of related e-mail messages or conversations, see [MS-ASCON].



Copyright © 2010 Microsoft Corporation.

For details about how outbound Short Message Service (SMS) e-mail messages are sent from mobile devices, see [MS-ASMS].

## 1.5 Prerequisites/Preconditions

This protocol assumes that authentication has been performed by the underlying protocols.

## 1.6 Applicability Statement

This protocol is applicable in scenarios where a client has to synchronize its messages and files with a server.

## 1.7 Versioning and Capability Negotiation

None.

## 1.8 Vendor-Extensible Fields

None.

## 1.9 Standards Assignments

None.



Copyright © 2010 Microsoft Corporation.

# 2 Messages

## 2.1 Transport

This protocol consists of a series of XML elements contained in request or response messages between a client and server. The XML block containing the command and parameter elements is transmitted in either the request body of a request, or in the response body of a response. The request body and request response are always preceded by the HTTP header, as specified in [MS-ASHTTP].

All command messages use **Wireless Application Protocol (WAP) Binary XML (WBXML)**, except for the **Autodiscover** command, which uses plain XML. For more details about WBXML, see [MS-ASWBXML].

## 2.2 Message Syntax

### 2.2.1 Namespaces

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

| Prefix | Namespace URI | Reference |
|---|---|---|
| airsync | AirSync | |
| airsyncbase | AirSyncBase | [MS-ASAIRS] |
| autodiscover | http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006 | |
| calendar | Calendar | [MS-ASCAL] |
| composemail | ComposeMail | |
| contacts | Contacts | [MS-ASCNTC] |
| contacts2 | Contacts2 | [MS-ASCNTC] |
| documentlibrary | DocumentLibrary | [MS-ASDOC] |
| email | Email | [MS-ASEMAIL] |
| email2 | Email2 | [MS-ASEMAIL] |
| folderhierarchy | FolderHierarchy | |
| gal | GAL | |
| getitemestimate | GetItemEstimate | |
| itemoperations | ItemOperations | |



Copyright © 2010 Microsoft Corporation.

| Prefix | Namespace URI | Reference |
|---|---|---|
| meetingresponse | MeetingResponse | |
| move | Move | |
| notes | Notes | [MS-ASNOTE] |
| ping | Ping | |
| provision | Provision | [MS-ASPROV] |
| resolverecipients | ResolveRecipients | |
| rm | RightsManagement | [MS-ASRM] |
| search | Search | |
| settings | Settings | |
| tasks | Tasks | [MS-ASTASK] |
| validatecert | ValidateCert | |
| xs | http://www.w3.org/2001/XMLSchema | [XMLSCHEMA1] |

### 2.2.2 Commands

#### 2.2.2.1 Autodiscover

The **Autodiscover** command facilitates the discovery of core account configuration information by using the user's **Simple Mail Transfer Protocol (SMTP)** address as the primary input. For details about the **Autodiscover** service, see [MSFT-AUTODISCOVER]. For more details about the Autodiscover HTTP Service, see [MS-OXDSCLI].

The **Autodiscover** command request and response messages are sent in XML format, not WBXML format.

When sending an **Autodiscover** command request, the Content-Type header value MUST be set to text/xml.<1> For more details about the Content-Type header, see [MS-ASHTTP] section 2.2.1.1.2.2.

The client SHOULD use the **Autodiscover** command as an initial response to common HTTP errors. Common HTTP errors are specified in [MS-ASHTTP] section 2.2.2.1.1. **Autodiscover** has the ability to retrieve an updated **Uniform Resource Locator (URL)** when a mailbox has been moved, a user is trying to connect to a server that cannot access the user's mailbox, or when there is a more efficient server to use to reach the user's mailbox.

After a successful **Autodiscover** command response, the client sends an **Options** command to the server identified in the **Autodiscover** command response. The **Options** command returns the newly supported protocol versions and commands if they changed due to the **Autodiscover** command.



Copyright © 2010 Microsoft Corporation.

The http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006 namespace is the primary namespace for the **Autodiscover** command and this section of the specification. Elements referenced in this section that are not defined in the http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006 namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.1.1 Request

The following code shows the **XML schema definition (XSD)** [XMLSCHEMA1] for the **Autodiscover** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns="http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006"
xmlns:mstns="http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2006"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/200
6"
elementFormDefault="unqualified" id="requestschema2006">
     <xs:element name="Autodiscover">
          <xs:complexType>
               <xs:sequence>
                    <xs:element name="Request" type="RequestType"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
     <xs:complexType name="RequestType">
          <xs:sequence>
               <xs:sequence>
                    <xs:element name="EMailAddress" type="xs:string"/>
                    <xs:element name="AcceptableResponseSchema" type="xs:string"/>
               </xs:sequence>
          </xs:sequence>
     </xs:complexType>
</xs:schema>
```

##### 2.2.2.1.1.1 Autodiscover

The <Autodiscover> element is the top-level element in the XML **stream**. It identifies the body of the HTTP **POST** as containing an **Autodiscover** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Request> (request only)<br><Response> (response only) | Container | 1 (required) |

###### 2.2.2.1.1.1.1 Request

The <Request> element contains the **Autodiscover** command request parameters.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Autodiscover> | <EMailAddress> (request only)<br><AcceptableResponseSchema> (request only) | **Container** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

When more than one <Request> elements are present in an **Autodiscover** request, the server returns an <ErrorCode> value of 600.

#### 2.2.2.1.1.1.1 EMailAddress

The <EMailAddress> element contains the SMTP e-mail address of the user and is used to identify the user's mailbox in the network.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only)<br><User> (response only) | None | **String** | 1...1 (required) |

If the user has multiple addresses, then the primary e-mail address SHOULD be returned in the **Autodiscover** command response. This address can be the same as the e-mail address that was sent in the request. The client device records this address string for use in all additional communication.

#### 2.2.2.1.1.1.2 AcceptableResponseSchema

The <AcceptableResponseSchema> element indicates the schema in which the server MUST send the response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only)<br><Response> (response only) | None | **String** | 1...1 (required) |

The namespace MUST be "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006".

### 2.2.2.1.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **Autodiscover** command response.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
xmlns="http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/2006a"

xmlns:mstns="http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/2006a"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/20
06a"
   elementFormDefault="unqualified" id="responseschema2006">
   <xs:element name="Autodiscover">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Response">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Culture" type="xs:string" minOccurs="0"/>
                     <xs:element name="User" minOccurs="0">
                        <xs:complexType>
                           <xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
                              <xs:element name="DisplayName" minOccurs="0"/>
                              <xs:element name="EMailAddress" minOccurs="0"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Action" minOccurs="0">
                        <xs:complexType>
                           <xs:all>
                              <xs:element name="Redirect" type="xs:string" minOccurs="0"/>
                              <xs:element name="Settings" minOccurs="0">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Server" maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Type" type="xs:string"
minOccurs="0"/>
                                                <xs:element name="Url" type="xs:string"
minOccurs="0"/>
                                                <xs:element name="Name" type="xs:string"
minOccurs="0"/>
                                                <xs:element name="ServerData"
type="xs:string" minOccurs="0"/>
                                             </xs:sequence>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="Error" minOccurs="0">
                                 <xs:complexType>
                                    <xs:all>
                                       <xs:element name="Status" type="xs:string"
minOccurs="0"/>
                                       <xs:element name="Message" type="xs:string"
minOccurs="0"/>
                                       <xs:element name="DebugData" type="xs:string"
minOccurs="0"/>
                                       <xs:element name="ErrorCode" type="xs:integer"
minOccurs="0"/>
                                    </xs:all>
                                 </xs:complexType>
                              </xs:element>
                           </xs:all>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Error" minOccurs="0">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="ErrorCode" type="xs:integer" minOccurs="0"/>
                              <xs:element name="Message" type="xs:string" minOccurs="0"/>
                              <xs:element name="DebugData" type="xs:string" minOccurs="0"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
```



Copyright © 2010 Microsoft Corporation.

```
    </xs:element>
  </xs:schema>
```

#### 2.2.2.1.2.1 Autodiscover

The <Autodiscover> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing an **Autodiscover** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Request> (request only)<br><Response> (response only) | Container | 1 (required) |

##### 2.2.2.1.2.1.1 Response

The <Response> element contains the **Autodiscover** command response parameters.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Autodiscover> (response only) | <Culture> (response only)<br><User> (response only)<br><Action> (response only)<br><Error> (response only) | **Container** | 1...1 (required) |

If an error occurs in the **Autodiscover** command framework that hosts the Autodiscovery implementation, then the <Response> element MUST have an <Error> child node.

###### 2.2.2.1.2.1.1.1 Culture

The <Culture> element specifies the client culture, which is used to localize error messages.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> | None | **String** | 0...1 (optional) |

The string MUST be of the form "en:us".<2>

###### 2.2.2.1.2.1.1.2 User

The <User> element encapsulates information about the user to whom this response element relates.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | <DisplayName> (response only)<br><EMailAddress> (response only) | **Container** | 1...1 (required) |

###### 2.2.2.1.2.1.1.2.1 DisplayName

The <DisplayName> element contains the user's display name in the directory service.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <User> (response only) | None | **String** | 0...1 (optional) |

The client can choose to display or store this value on the device.

#### 2.2.2.1.2.1.1.2.2 EMailAddress

The <EMailAddress> element contains the SMTP e-mail address of the user and is used to identify the user's mailbox in the network.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only)<br><User> (response only) | None | **String** | 1...1 (required) |

If the user has multiple addresses, then the primary e-mail address is returned in the **Autodiscover** command response. The primary e-mail address is the address that appears on the From line when a user sends an e-mail message. This address can be the same as the e-mail address that was sent in the request. The client device SHOULD record the address returned by the **Autodiscover** command response and SHOULD use this string for all additional communication.

#### 2.2.2.1.2.1.1.3 Action

The <Action> element encapsulates the server action type for this request, which can be one of the following: <Redirect>, <Settings>, or <Error>.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | <Redirect> (response only)<br><Settings> (response only)<br><Error> (response only) | **Container** | 0...1 (optional) |

#### 2.2.2.1.2.1.1.3.1 Redirect

The <Redirect> element specifies the SMTP address of the requested user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Action> (response only) | None | **String** | 0...1 (optional) |

The <Redirect> element is an optional child of the <Action> element in the **Autodiscover** response message. The client device uses the **domain** part of the address to send a new **Autodiscover** command request.

#### 2.2.2.1.2.1.1.3.2 Settings

The <Settings> element contains the settings for the specified user or schema.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Action> (response only) | <Server> | **Container** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.1.2.1.1.3.2.1 Server

The <Server> element encapsulates settings that apply to a particular server in the **Autodiscover** command response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> (response only) | <Type> (response only)<br><URL> (response only)<br><Name> (response only)<br><ServerData> (response only) | **Container** | 1...N (required) |

#### 2.2.2.1.2.1.1.3.2.1.1 Type

The <Type> element specifies the server type.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Server> (response only) | None | **String** | 0...1 (optional) |

The following are the valid values for the <Type> element:

- *MobileSync*. Indicates that the URL that is returned by the URL element can be accessed by clients.
- *CertEnroll*. Indicates that the URL that is returned by the URL element can be accessed by clients that have a valid certificate over a **Secure Sockets Layer (SSL)**.

If the server supports both *MobileSync* and **CertEnroll**, the response buffer includes multiple <Server> elements that contain a URL value for each <Type> value.

#### 2.2.2.1.2.1.1.3.2.1.2 Url

The <URL> element contains a URL string that conveys the protocol, port, resource location, and other information.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Server> (response only) | None | **String** | 0...1 (optional) |

The URL element is a child of the <Server> element. The value is a URL string that conveys the protocol, port, resource location, and other information.

#### 2.2.2.1.2.1.1.3.2.1.3 Name

The <Name> element specifies a URL if the **Type** element is set to *MobileSync*.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Server> (response only) | None | **String** | 0...1 (optional) |

If the <Type> element value is *MobileSync*, then the <Name> element specifies the URL that conveys the protocol. If the <Type> element value is *CertEnroll*, then the <Name> value is NULL.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.1.2.1.1.3.2.1.4 ServerData

The <ServerData> element contains the template name for the client certificate.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Server> (response only) | None | **String** | 0...1 (optional) |

The <ServerData> element is a string that is present only when the <Type> element value is set to **CertEnroll**.

#### 2.2.2.1.2.1.1.3.3 Error

The <Error> element contains the error that was encountered while processing the request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Action> (response only) | <Status> (response only)<br><Message> (response only)<br><DebugData> (response only)<br><ErrorCode> (response only) | **Container** | 0...1 (optional) |

#### 2.2.2.1.2.1.1.3.3.1 Status

The <Status> element provides a status code that corresponds to the error.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **Integer** | 0...1 (optional) |

The following table specifies valid values for the <Status> element in the context of the <Settings> element.

| Value | Meaning |
|---|---|
| 1 | Success. Because the <Status> element is only returned when the command encounters an error, the success status code is never included in a response message. |
| 2 | Protocol error |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

The client device can implement custom recovery logic pertaining to the status code. If an unknown status code is returned to the client, the client SHOULD have logic in place to handle the error by sending an error message to the user, resending the command with new settings, or custom logic.

#### 2.2.2.1.2.1.1.3.3.2 Message

The <Message> element contains the error string localized using the <Culture> specified in the <Response> element, enabling the client to display error status to the end-user.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **String** | 0...1 (optional) |

#### 2.2.2.1.2.1.1.3.3.3 DebugData

The <DebugData> element represents more information about the failure that can help the system administrator debug the source of the problem.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **String** | 0...1 (optional) |

This element is not intended for use by developers debugging their own data.

#### 2.2.2.1.2.1.1.3.3.4 ErrorCode

The <ErrorCode> element contains an error number that indicates the cause of the error.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **Integer** | 0...1 (optional) |

If the provider cannot be found, or if the <AcceptableResponseSchema> cannot be matched, then the <ErrorCode> is included in the command response. A value of 600 means an invalid request was sent to the server, a value of 601 means that a provider could not be found to handle the <AcceptableResponseSchema> that was specified.

### 2.2.2.1.2.1.1.4 Error

The <Error> element contains the error that was encountered while processing the request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | <ErrorCode> (response only)<br><Message> (response only)<br><DebugData> (response only) | **Container** | 0...1 (optional) |

#### 2.2.2.1.2.1.1.4.1 ErrorCode

The <ErrorCode> element contains an error number that indicates the cause of the error.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **Integer** | 0...1 (optional) |

If the provider cannot be found, or if the <AcceptableResponseSchema> cannot be matched, then the <ErrorCode> is included in the command response. A value of 600 means an invalid request was sent to the server, a value of 601 means that a provider could not be found to handle the <AcceptableResponseSchema> that was specified.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.1.2.1.1.4.2 Message

The <Message> element contains the error string localized using the <Culture> specified in the <Response> element, enabling the client to display error status to the end-user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **String** | 0...1 (optional) |

#### 2.2.2.1.2.1.1.4.3 DebugData

The <DebugData> element represents more information about the failure that can help the system administrator debug the source of the problem.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Error> (response only) | None | **String** | 0...1 (optional) |

This element is not intended for use by developers debugging their own data.

### 2.2.2.2 FolderCreate

The **FolderCreate** command creates a new folder as a child of the specified parent folder. A parent ID of 0 signifies the mailbox **root folder**.

The **FolderCreate** command cannot be used to create a **recipient information cache** or a subfolder of a recipient information cache.

The FolderHierarchy namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the FolderHierarchy namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.2.1 Request

The server that is implementing this protocol enforces the following XSD [XMLSCHEMA1] when processing protocol requests.

Requests that do not adhere to the schema result in the return of a status 10 to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderCreate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ParentId">
                    <xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="DisplayName">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="256"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Type" type="xs:unsignedByte" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.2.1.1 FolderCreate

The <FolderCreate> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **FolderCreate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ParentId> (request only)<br><DisplayName> (request only)<br><Type> (request only)<br><ServerId> (response only)<br><Status> (response only) | **Container** | 1 (required) |

##### 2.2.2.2.1.1.1 SyncKey

The <SyncKey> element specified in the **FolderCreate** command request represents the synchronization state of a **collection**. After a successful **FolderCreate** command, the server sends a synchronization key to the client in a response. The client MUST store this key and send it back to the server the next time the folder hierarchy is synchronized or updated. The server checks the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> | None | **String** (Up to 64 characters) | Request: 1 (required)<br>Response: 0...1 (optional) |

The client MUST store the synchronization key as an opaque string of up to 64 characters.

The <SyncKey> element is returned if the **FolderCreate** command request was successful and the element is not returned if the **FolderCreate** command request fails.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.2.1.1.2 ParentId

The <ParentId> element specifies the server ID of the parent folder and is used in **FolderCreate** command requests only. The server ID of the parent folder is obtained from the <ServerId> element of a previous **FolderSync** command. A parent ID of 0 signifies the mailbox root folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

#### 2.2.2.2.1.1.3 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> (request only) | None | **String** (Between 1 and 256 characters) | 1 (required) |

#### 2.2.2.2.1.1.4 Type

The <Type> element specifies the type of the folder to be created.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> (request only) | None | **Unsigned byte** | 1 (required) |

The folder type values are listed in the following table. Folder types 2–11 and 19 are reserved for default folder types.

| Type | Definition |
|---|---|
| 1 | User-created folder (generic) |
| 12 | User-created mail folder |
| 13 | User-created **Calendar folder** |
| 14 | User-created Contacts folder |
| 15 | User-created Tasks folder |
| 16 | User-created **Journal** folder |
| 17 | User-created Notes folder |

### 2.2.2.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **FolderCreate** command response.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```



Copyright © 2010 Microsoft Corporation.

```
    <xs:element name="FolderCreate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element name="SyncKey" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                             <xs:maxLength value="64"/>
                             <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="ServerId" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                             <xs:maxLength value="64"/>
                             <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.2.2.1 FolderCreate

The <FolderCreate> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** response as containing a **FolderCreate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ParentId> (request only)<br><DisplayName> (request only)<br><Type> (request only)<br><ServerId> (response only)<br><Status> (response only) | **Container** | 1 (required) |

##### 2.2.2.2.2.1.1 Status

The <Status> element indicates in the **FolderCreate** command response the success or failure of a **FolderCreate** command request. If the command failed, the <Status> element contains a code indicating the type of failure. The values are summarized in the following table.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> (response only) | None | **Unsigned byte** | 1 (required) |

The following table lists the <Status> codes for the **FolderCreate** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | A folder that has this name already exists. | The parent folder already contains a folder that has this name. | Item | Prompt user to supply a unique name. |
| 3 | The specified folder is a special system folder. | The specified folder is a special system folder, like the **Inbox**, **Outbox**, Contacts, **Recipient** information, or **Drafts folders**, and cannot be created by the client. | Item | Create a folder with a different type. |
| 5 | The specified parent folder was not found. | The parent folder does not exist on the server, possibly because it has been deleted or renamed. | Item | Issue a **FolderSync** command for the new hierarchy and prompt the user for a new parent folder. |
| 6 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **FolderSync** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0). |
| 8 | The request timed out. | The server took too long to respond to the request. | Global | Retry. |
| 9 | Synchronization key mismatch or invalid synchronization key. | The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server. | Global | Delete folders added since last synchronization and return to synchronization key to zero (0). |
| 10 | Malformed request. | The client sent a **FolderCreate** command that contains a semantic error. | Global | Fix bug in client code. Double-check the request for accuracy. |
| 11 | An unknown error occurred. | Unknown. | Global. | No solution. |
| 12 | Code unknown. | Unusual back-end issue. | Global | No solution. |

#### 2.2.2.2.2.1.2 SyncKey

The <SyncKey> element specified in the **FolderCreate** command response represents the synchronization state of a collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> | None | **String** (Up to 64 characters) | Request: 1 (required)<br>Response: 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

After a successful **FolderCreate** command, the server MUST send a synchronization key to the client in a response. If the **FolderCreate** command is not successful, the server MUST NOT return a <SyncKey> element.

The client MUST store this key and send it back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a Status value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

#### 2.2.2.2.2.1.3 ServerId

The <ServerId> element uniquely identifies a new folder on a server. The <ServerId> of the new folder is returned to the client after a successful **FolderCreate** command request. The <ServerId> can also be used in the <ServerId> element of future **FolderDelete** and **FolderUpdate** command requests. The client MUST store the <ServerId> for each folder and MUST be able to locate a folder on the client given a <ServerId>.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderCreate> (response only) | None | **String** (Up to 64 characters) | 0...1 (optional) |

The <ServerId> element MUST be returned if the **FolderCreate** command request was successful and the element MUST NOT be returned if the **FolderCreate** command request fails.

### 2.2.2.3 FolderDelete

The **FolderDelete** command deletes a folder from the server. The <ServerId> of the folder is passed to the server in the **FolderDelete** command request, which deletes the collection with the matching identifier. The server then sends a response indicating the status of the deletion.

The **FolderDelete** command cannot be used to delete a recipient information cache. Attempting to delete a recipient information cache using this command results in a <Status> value of 3.

The FolderHierarchy namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the FolderHierarchy namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.3.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **FolderDelete** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderDelete">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ServerId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.3.1.1 FolderDelete

The <FolderDelete> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **FolderDelete** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ServerId> (request only)<br><Status> (response only) | **Container** | 1 (required) |

##### 2.2.2.3.1.1.1 SyncKey

The <SyncKey> element represents the synchronization state of a folder hierarchy.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderDelete> | None | **String** (Up to 64 characters) | Request: 1 (required)<br>Response: 0...1 (optional) |

After a successful **FolderDelete** command request, the server MUST send a synchronization key to the client in the response. If the **FolderDelete** command request is unsuccessful, the server MUST NOT return a <SyncKey> element.

The client MUST store this key and send it back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

##### 2.2.2.3.1.1.2 ServerId

The <ServerId> element specifies the folder on the server to be deleted, and it is a unique identifier assigned by the server to each folder that can be synchronized.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderDelete> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The server ID of the folder to be deleted is returned to the client in the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The client MUST store the server ID for each folder and MUST be able to locate a folder given a server ID.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

#### 2.2.2.3.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **FolderDelete** command response.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderDelete">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" name="SyncKey">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

##### 2.2.2.3.2.1 FolderDelete

The <FolderDelete> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** response as containing a **FolderDelete** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ServerId> (request only)<br><Status> (response only) | **Container** | 1 (required) |

###### 2.2.2.3.2.1.1 Status

The <Status> element indicates the success or failure of the **FolderDelete** command request. If the command failed, the <Status> element in the server response contains a code indicating the type of failure.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderDelete> (response) | None | **Unsigned byte** | 1 (required) |

The following table lists the <Status> codes for the **FolderDelete** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 3 | The specified folder is a special system folder, such as the Inbox, Outbox, Contacts, Recipient information, or Drafts folders, and cannot be deleted by the client. | The client specified a **special folder** in a **FolderDelete** command request. special folders cannot be deleted. | Item | None. |
| 4 | The specified folder does not exist. | The client specified a nonexistent folder in a **FolderDelete** command request. | Item | Issue a **FolderSync** command for the new hierarchy. |
| 6 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **FolderDelete** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0). |
| 8 | The request timed out. | The server took too long to respond to the request. | Global | Retry. |
| 9 | Synchronization key mismatch or invalid synchronization key. | The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server. | Global | Issue a **FolderSync** command request with a synchronization key of zero (0). |
| 10 | Incorrectly formatted request. | The client sent a **FolderCreate** command request that contains a semantic or syntactic error. | Global | Fix bug in client code. Double-check the request for accuracy. |
| 11 | An unknown error occurred. | Unusual back-end issue. | Global | No solution. |

#### 2.2.2.3.2.1.2 SyncKey

The <SyncKey> element is used by the server to mark the synchronization state of a folder hierarchy.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderDelete> | None | **String** (Up to 64 characters) | Request: 1 (required)<br>Response: 0...1 |

After a successful **FolderDelete** command, the server MUST send a synchronization key to the client in a response. If the **FolderDelete** command is not successful, the server MUST NOT return a <SyncKey> element.

The client MUST store this key and send it back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

### 2.2.2.4 FolderSync

The **FolderSync** command synchronizes the collection hierarchy but does not synchronize the items in the collections themselves.

The FolderHierarchy namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the FolderHierarchy namespace use the namespace prefixes defined in section 2.2.1.

This command works similarly to the **Sync** command. An initial **FolderSync** command with a synchronization key of 0 (value of 0 in <SyncKey> element) is required in order to obtain the list of folders and the synchronization key associated with that list. The synchronization key MUST be returned in the <SyncKey> element of the response. This synchronization key MUST be used in subsequent **FolderSync** commands to obtain folder hierarchy changes.

Unlike a **Sync** request, there is no <GetChanges> element submitted in the **FolderSync** request to get changes from the server. All folders MUST be returned to the client when initial folder synchronization is done with a synchronization key of 0.

#### 2.2.2.4.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **FolderSync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderSync">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                 <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="64"/>
                    </xs:restriction>
                 </xs:simpleType>
                </xs:element>
            </xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.4.1.1 FolderSync

The <FolderSync> element is the top-level element in the XML stream. It indicates that the body of the HTTP POST contains a **FolderSync** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><Status> (response only)<br><Changes> (response only) | **Container** | 1 (required) |

##### 2.2.2.4.1.1.1 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderSync> | None | **String** (Up to 64 characters) | 1 (required) |

After successful folder synchronization, the server MUST send a synchronization key to the client. The client MUST store this key and send the key back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

If a synchronization error occurs, and the **FolderSync** response contains status code 9 (see section 2.2.2.4.2.1.1), then the client MUST restart the synchronization process with a synchronization key of 0. The client's folder hierarchy list MUST be rebuilt and any changes that existed on the client that had not been propagated to the server prior to the error SHOULD be sent after the **FolderSync** is complete.

### 2.2.2.4.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **FolderSync** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="FolderHierarchy:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="FolderSync">
          <xs:complexType>
               <xs:sequence>
                    <xs:element name="Status" type="xs:unsignedByte" />
                    <xs:element minOccurs="0" name="SyncKey" >
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:maxLength value="64"/>
                                <xs:minLength value="1"/>
```



Copyright © 2010 Microsoft Corporation.

```
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element minOccurs="0" name="Changes">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="Count" minOccurs="0"
type="xs:unsignedInt" />
                                <xs:element minOccurs="0" maxOccurs="unbounded"
name="Update">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="ServerId" >
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                        <xs:minLength value="1"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                            <xs:element name="ParentId" >
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                        <xs:minLength value="1"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                            <xs:element name="DisplayName"
type="xs:string" />
                                            <xs:element name="Type" type="xs:integer"
/>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                                <xs:element minOccurs="0" maxOccurs="unbounded"
name="Delete">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="ServerId" >
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                        <xs:minLength value="1"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                                <xs:element minOccurs="0" maxOccurs="unbounded"
name="Add">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="ServerId" >
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                        <xs:minLength value="1"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                            <xs:element name="ParentId" >
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                        <xs:minLength value="1"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                            <xs:element name="DisplayName"
type="xs:string" />

                                            <xs:element name="Type" type="xs:integer"
/>
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

#### 2.2.2.4.2.1 FolderSync

The <FolderSync> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **FolderSync** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><Status> (response only)<br><Changes> (response only) | **Container** | 1 (required) |

##### 2.2.2.4.2.1.1 Status

The <Status> element indicates the success or failure of a **FolderSync** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderSync> (response only) | None | **Unsigned byte** (See values in the following table) | 1 (required) |

If the command fails, the <Status> element contains a code that indicates the type of failure. The <Status> element is global for all collections. If one collection fails, a failure status MUST be returned for all collections.

The following table lists the <Status> codes for the **FolderSync** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success | Server successfully completed command. | Global | None. |
| 6 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **FolderSync** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0). |
| 8 | The request timed out. | The server took too long to respond to the request. | Global | Retry. |
| 9 | Synchronization key mismatch or invalid synchronization key. | The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server. | Global | Delete items added since last synchronization and return to synchronization key zero (0). |
| 10 | Incorrectly formatted request. | The client sent a **FolderSync** command request that contains a semantic or syntactic error. | Global | Fix bug in client code. Double-check the request for accuracy. |
| 11 | An unknown error occurred. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **FolderSync** command request. If continued attempts to synchronization fail, consider returning to synchronization key zero (0). |
| 12 | Code unknown. | Unusual back-end issue. | Global | No solution. |

#### 2.2.2.4.2.1.2 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderSync> | None | **String** (Up to 64 characters) | 0…1 (optional) |

After a successful folder synchronization, the server MUST send a synchronization key to the client. The client MUST store this key and send the key back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

If a synchronization error occurs, where the **FolderSync** response contains status code 9 (see section 2.2.2.4.2.1.1), the client MUST restart the synchronization process with a synchronization key of 0. The client's folder hierarchy list MUST be rebuilt and any changes that existed on the client that had not been propagated to the server prior to the error SHOULD be sent after the **FolderSync** is complete.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.4.2.1.3 Changes

The <Changes> element is a container for changes to the folder hierarchy. It is used in the **FolderSync** command response to update the client with folder additions, deletions, and updates on the server.

The server SHOULD maintain the same set of folder data being returned across synchronization key 0, in terms of <ServerId> and <DisplayName> mapping. While the server SHOULD maintain this mapping, clients MUST be able to handle having the server return a completely different set of mapped data.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderSync> (response only) | <Count> (response only)<br><Update> (response only)<br><Delete> (response only)<br><Add> (response only) | **Container** | 0…1 (optional) |

#### 2.2.2.4.2.1.3.1 Count

The <Count> element is used in the **FolderSync** command response to list the number of added, deleted, and updated folders on the server since the last folder synchronization. These changes are listed in the <Changes> element. If there are no changes since the last folder synchronization, a <Count> of 0 is returned.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Changes> (response only) | None | **Unsigned Integer** | 0…1 (optional) |

#### 2.2.2.4.2.1.3.2 Update

The <Update> element is used in a **FolderSync** command response to identify a folder on the server that has been updated (renamed or moved).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Changes> (response only) | <ServerId> (response only)<br><ParentId> (response only)<br><DisplayName> (response only)<br><Type> (response only) | **Container** | 0...N (optional) |

The child elements of the <Update> element identify the server ID of the folder that was updated, the server ID of its parent folder, the new display name of the updated folder, and the folder type.

#### 2.2.2.4.2.1.3.2.1 ServerId

The <ServerId> element specifies the server-unique identifier for a folder on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Update> (response only) | None | **String** (Up to 64 characters) | 1 (required) |



Copyright © 2010 Microsoft Corporation.

The <ServerId> element is used to identify folders that have been updated on the server in the **FolderSync** command response.

The client MUST store the server ID as an opaque string of up to 64 characters.

Each <Update> element included in a **FolderSync** response MUST contain one <ServerId> element.

#### 2.2.2.4.2.1.3.2.2 ParentId

The <ParentId> element specifies the server ID of the parent of the folder on the server that has been updated.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Update> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the parent ID as an opaque string of up to 64 characters.

Each <Update> element included in a **FolderSync** response MUST contain one <ParentId> element.

#### 2.2.2.4.2.1.3.2.3 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Update> (response only) | None | **String** | 1 |

One <DisplayName> element is used in each <Update> element included in a **FolderSync** response when a folder has been updated on the server. Subfolder display names MUST be unique within a folder.

#### 2.2.2.4.2.1.3.2.4 Type

The <Type> element specifies the type of the folder that was updated (renamed or moved) on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Update> (response only) | None | **Integer** | 1 (required, as a child of <Add> and <Update>) |

Each <Update> element included in a **FolderSync** response MUST contain one <Type> element.

The folder type values are listed in the following table.

| Value | Meaning |
|---|---|
| 1 | User-created folder (generic) |
| 2 | Default Inbox folder |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning |
|---|---|
| 3 | Default Drafts folder |
| 4 | Default Deleted Items folder |
| 5 | Default **Sent Items folder** |
| 6 | Default Outbox folder |
| 7 | Default Tasks folder |
| 8 | Default Calendar folder |
| 9 | Default Contacts folder |
| 10 | Default Notes folder |
| 11 | Default Journal folder |
| 12 | User-created Mail folder |
| 13 | User-created Calendar folder |
| 14 | User-created Contacts folder |
| 15 | User-created Tasks folder |
| 16 | User-created journal folder |
| 17 | User-created Notes folder |
| 18 | Unknown folder type |
| 19 | Recipient information cache |

#### 2.2.2.4.2.1.3.3 Delete

The <Delete> element is used in the **FolderSync** command response to specify that a folder on the server was deleted since the last folder synchronization.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Changes> (response only) | <ServerId> (response only) | **Container** | 0...N (optional) |

##### 2.2.2.4.2.1.3.3.1 ServerId

The <ServerId> element specifies the server-unique identifier for a folder on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Delete> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

The <ServerId> element is used to identify folders that have been deleted on the server in the **FolderSync** command response.

The client MUST store the server ID as an opaque string of up to 64 characters.



Copyright © 2010 Microsoft Corporation.

Each <Delete> element included in a **FolderSync** response MUST contain one <ServerId> element.

#### 2.2.2.4.2.1.3.4 Add

The <Add> element is used in a **FolderSync** command response to create a new folder on the client. Child elements of the <Add> element specify the server ID of the folder, the server ID of the parent folder, the display name of the folder, and the type of folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Changes> (response only) | <ServerId> (response only)<br><ParentId> (response only)<br><DisplayName> (response only)<br><Type> (response only) | **Container** | 0...N (optional) |

#### 2.2.2.4.2.1.3.4.1 ServerId

The <ServerId> element specifies the server-unique identifier for a folder on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

The <ServerId> element is used to identify folders that have been added to the server in the **FolderSync** command response.

The client MUST store the server ID as an opaque string of up to 64 characters.

Each <Add> element included in a **FolderSync** response MUST contain one <ServerId> element.

#### 2.2.2.4.2.1.3.4.2 ParentId

The <ParentId> element specifies the server ID of the parent of the folder on the server that has been added.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **String** (Up to 64 characters) | 1 (required, as a child of <Add> and <Update>) |

The client MUST store the parent ID as an opaque string of up to 64 characters.

Each <Add> element included in a **FolderSync** response MUST contain one <ParentId> element.

#### 2.2.2.4.2.1.3.4.3 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **String** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

One <DisplayName> element is used in each <Add> element included in a **FolderSync** response when a folder has been added on the server. Subfolder display names MUST be unique within a folder.

#### 2.2.2.4.2.1.3.4.4 Type

The <Type> element specifies the type of the folder that was added to the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **Integer** | 1 (required) |

Each <Add> element included in a **FolderSync** response MUST contain one <Type> element.

The folder type values are listed in the following table.

| Value | Meaning |
|---|---|
| 1 | User-created folder (generic) |
| 2 | Default Inbox folder |
| 3 | Default Drafts folder |
| 4 | Default Deleted Items folder |
| 5 | Default Sent Items folder |
| 6 | Default Outbox folder |
| 7 | Default Tasks folder |
| 8 | Default Calendar folder |
| 9 | Default Contacts folder |
| 10 | Default Notes folder |
| 11 | Default Journal folder |
| 12 | User-created Mail folder |
| 13 | User-created Calendar folder |
| 14 | User-created Contacts folder |
| 15 | User-created Tasks folder |
| 16 | User-created journal folder |
| 17 | User-created Notes folder |
| 18 | Unknown folder type |
| 19 | Recipient information cache |



Copyright © 2010 Microsoft Corporation.

### 2.2.2.5 FolderUpdate

The **FolderUpdate** command moves a folder from one location to another on the server. The command is also used to rename a folder.

The **FolderUpdate** command cannot be used to update a recipient information cache, or to move a folder under the recipient information cache. Attempting to update a recipient information cache using this command results in a <Status> value of 3.

The FolderHierarchy namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the FolderHierarchy namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.5.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **FolderUpdate** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderUpdate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ServerId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ParentId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="DisplayName">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="256"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
        </xs:complexType>
      </xs:element>
    </xs:schema>
```

#### 2.2.2.5.1.1 FolderUpdate

The <FolderUpdate> element is the top-level element in the XML stream. It indicates that the body of the HTTP POST contains a **FolderUpdate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ServerId> (request only)<br><ParentId> (request only)<br><DisplayName> (request only)<br><Status> (response only) | **Container** | 1 (required) |

Including the <Status> element in a **FolderUpdate** request results in a <Status> value of 10 being returned in the response.

##### 2.2.2.5.1.1.1 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> | None | **String** (Up to 64 characters) | 1 (required) |

The server returns a <Status> value of 10 if the <SyncKey> value is not included in the **FolderUpdate** request.

After a successful **FolderUpdate** command, the server MUST send a new synchronization key to the client. If the **FolderUpdate** command was not successful, the server MUST NOT return a <SyncKey> element.

The client MUST store this key and send the key back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

##### 2.2.2.5.1.1.2 ServerId

The <ServerId> element identifies the folder on the server to be renamed or moved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The server ID is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The <ServerId> specifies a unique identifier assigned by the server to each object that



Copyright © 2010 Microsoft Corporation.

can be synchronized. The client MUST store the <ServerId> for each folder and MUST be able to locate a folder given a <ServerId>.

The client MUST store the <ServerId> as an opaque string of up to 64 characters.

#### 2.2.2.5.1.1.3 ParentId

The <ParentId> element specifies the server ID of the parent of the folder to be renamed or the destination folder of the folder to be moved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The <ParentId> is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The client MUST store the <ParentId> as an opaque string of up to 64 characters.

A <ParentId> of 0 signifies the mailbox root folder.

#### 2.2.2.5.1.1.4 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> (request only) | None | **String** (up to 256 characters) | 1 (required) |

### 2.2.2.5.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **FolderUpdate** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="FolderHierarchy:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderUpdate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" name="SyncKey">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                            <xs:minLength value="1"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.5.2.1 FolderUpdate

The <FolderUpdate> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **FolderUpdate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <SyncKey><br><ServerId> (request only)<br><ParentId> (request only)<br><DisplayName> (request only)<br><Status> (response only) | **Container** | 1 (required) |

##### 2.2.2.5.2.1.1 Status

The <Status> element indicates the success or failure of a **FolderUpdate** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> (response only) | None | **Unsigned byte** (See values in the following table) | 1 (required) |

If the command fails, the <Status> element contains a code that indicates the type of failure.

The following table lists the <Status> codes for the **FolderUpdate** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Code | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | A folder with that name already exists or the specified folder is a special folder. | A folder with that name already exists or the specified folder is a special folder, such as the Inbox, Outbox, Contacts, or Drafts folders. Special folders cannot be updated. | Item | None. |
| 3 | The specified folder is Recipient information folder, which cannot be updated by the client. | The client specified the Recipient information folder, which is a special folder. Special folders cannot be updated. | Item | None. |
| 4 | The specified folder does not exist. | Client specified a nonexistent folder in a **FolderUpdate** command request. | Item | Issue a **FolderSync** command for the new hierarchy. |
| 5 | The specified parent folder was not found. | Client specified a nonexistent folder in a **FolderUpdate** command request. | Item | Issue a **FolderSync** command for the new hierarchy. |



Copyright © 2010 Microsoft Corporation.

| Code | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 6 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **FolderUpdate** command request. If continued attempts to synchronization fail, consider returning to synchronization key 0. |
| 8 | The request timed out. | The server took too long to respond to the request. | Global | Retry. |
| 9 | Synchronization key mismatch or invalid synchronization key. | The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server. | Global | Issue a **FolderSync** command request with a synchronization key of 0. |
| 10 | Incorrectly formatted request. | The client sent a **FolderUpdate** command request that contains a semantic error. | Global | Fix bug in client code. Double-check the request for accuracy. |
| 11 | An unknown error occurred. | Unusual back-end issue. | Global | No solution. |

#### 2.2.2.5.2.1.2 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <FolderUpdate> | None | **String** (Up to 64 characters) | 1 (required) |

After a successful **FolderUpdate** command, the server MUST send a new synchronization key to the client. If the **FolderUpdate** command was not successful, the server MUST NOT return a <SyncKey> element.

The client MUST store this key and send the key back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

### 2.2.2.6 GetAttachment

The **GetAttachment** command retrieves an e-mail attachment from the server.<3>

Attachments are not automatically included with e-mail messages in a synchronization; they are explicitly retrieved by using the **GetAttachment** command.

This command is issued within the HTTP **POST** command, and does not require any additional information in an XML body. The name of the attachment to be retrieved is specified in the *AttachmentName* command parameter. The *AttachmentName* parameter MUST be **base64 encoded**, as specified in [MS-ASHTTP] section 2.2.1.1.1.



Copyright © 2010 Microsoft Corporation.

The content of the attachment is returned in the response body with the content type being specified in the *Content-Type* header of the response. When the *Content-Type* header is missing, this indicates that the default encoding of 7-bit **ASCII** has been used.

If the **GetAttachment** command is used to retrieve an attachment that has been deleted on the server, a 500 status code is returned in the HTTP **POST** response.

#### 2.2.2.6.1 Request

No XML body is included in the **GetAttachment** command request.

#### 2.2.2.6.2 Response

The content of the attachment is returned in the response body with the content type being specified in the Content-Type header of the response. When the Content-Type header is missing, this indicates that the default encoding of 7-bit ASCII has been used.

No XML body is included in the **GetAttachment** command response.

### 2.2.2.7 GetItemEstimate

The **GetItemEstimate** command gets an estimate of the number of items in a collection or folder on the server that have to be synchronized.

The GetItemEstimate namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the GetItemEstimate namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.7.1 Request

The following code defines the XSD [XMLSCHEMA1] for the **GetItemEstimate** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="GetItemEstimate:"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:airsync="AirSync:"
   targetNamespace="GetItemEstimate:"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">
   <xs:import namespace="AirSync:"/>
   <xs:element name="GetItemEstimate">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Collections">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Collection" maxOccurs="300">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element ref="airsync:SyncKey"/>
                              <xs:element name="CollectionId">
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                       <xs:maxLength value="64"/>
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:element>
```



Copyright © 2010 Microsoft Corporation.

```
                                <xs:element ref="airsync:ConversationMode" minOccurs="0"/>
                                <xs:element ref="airsync:Options" minOccurs="0" maxOccurs="2"/>
                                <xs:sequence>
                                   <xs:element ref="airsync:Class" minOccurs="0"/>
                                   <xs:element ref="airsync:FilterType" minOccurs="0"/>
                                   <xs:element ref="airsync:MaxItems" minOccurs="0"/>
                                </xs:sequence>
                             </xs:sequence>
                          </xs:complexType>
                       </xs:element>
                    </xs:sequence>
                 </xs:complexType>
              </xs:element>
           </xs:sequence>
        </xs:complexType>
     </xs:element>
  </xs:schema>
```

#### 2.2.2.7.1.1 GetItemEstimate

The required element <GetItemEstimate> is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **GetItemEstimate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Collections> (request only)<br><Response> (response only) | **Container** | 1 (required) |

#### 2.2.2.7.1.1.1 Collections

The <Collections> element serves as a container for one to 300 <Collection> elements.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <GetItemEstimate> (request only) | <Collection> | **Container** | 1 (required) |

#### 2.2.2.7.1.1.1.1 Collection

The <Collection> element wraps elements that apply to a particular collection. A maximum of 300 <Collection> elements can be included in a single <Collections> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collections> (request only)<br><Response> (response only) | <airsync:SyncKey> (request only)<br><CollectionId><br><airsync:ConversationMode> (request only)<br><airsync:Options> (request only)<br><Estimate> (response only)<4> | **Container** | 1...300 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.7.1.1.1.1.1 airsync:SyncKey

The required element <airsync:SyncKey> represents the current state of a collection. The value of the element is examined by the server to determine the state of the synchronization process. The <airsync:SyncKey> element is the first child element of <Collection> in a **GetItemEstimate** command request.<5>

The <airsync:SyncKey> used within the <GetItemEstimate> requests is the same as the one returned within the **Sync** responses. The server does not update the <airsync:SyncKey> on **GetItemEstimate** requests. For more details about the <airsync:SyncKey> element, see section 2.2.2.19.1.2.1.1.1.The server checks the value of the key to verify that the value of the <airsync:SyncKey> provided in the request matches a <airsync:SyncKey> value on the server. The server MUST provide a <Status> value of 4 if the <airsync:SyncKey> value provided in the **GetItemEstimate** request does not match those expected within the next **Sync** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

#### 2.2.2.7.1.1.1.1.2 CollectionId

The <CollectionId> element specifies the server ID of the collection from which the item estimate is being obtained.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | **String** (Up to 64 characters) | 1...1 (required) |

The <CollectionId> value sent in the **GetItemEstimate** command request corresponds to the <ServerId> element value assigned to the folder containing the item to retrieve. The client SHOULD store <ServerId> values as they are returned by the server in **FolderSync** and **FolderCreate** command responses. The <CollectionId> element is used in both **GetItemEstimate** command requests and responses.

#### 2.2.2.7.1.1.1.1.3 airsync:ConversationMode

The optional element <airsync:ConversationMode><6> specifies whether to include items that are included within the **conversation** modality within the results of the **GetItemEstimate** response. A single conversation MAY span multiple collections and therefore <airsync:ConversationMode> is a child of the <Collection> element, rather than the <airsync:Options> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **Boolean** | 0...1 (optional) |

Setting the <airsync:ConversationMode> element to **FALSE** in a **GetItemEstimate** request results in an <Estimate> value that only includes items that meet the <airsync:FilterType> value. Setting <airsync:ConversationMode> to **TRUE** expands the result set to also include items with identical <email2:ConversationId> values to those in the <airsync:FilterType> result set. The <airsync:ConversationMode> value has no impact on items outside the collection specified by the <CollectionId>, the result set is always limited to items in the specified collection. The <airsync:ConversationMode> value only limits or expands the results determined by the <airsync:FilterType> value.



Copyright © 2010 Microsoft Corporation.

Specifying <airsync:ConversationMode> for collections that do not store e-mail results in an invalid XML error, status code 103.

The result of including a <airsync:ConversationMode> element as the child of a <airsync:ConversationMode> element is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.7.1.1.1.1.4 airsync:Options

The <airsync:Options><7> element is a container that encloses elements that filter the results of the **GetItemEstimate** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | <airsync:Class> (request only)<br><airsync:FilterType> (request only)<br><airsync:MaxItems> (request only) | **Container** | 0...2 (optional) |

This element is optional; however, when it is present, it MUST include at least one child element. The <airsync:Options> element appears only in requests to the server from the client. If the <airsync:Options> element is not included in a request, then the **GetItemEstimate** command will enumerate all of the items within the collection, without any filter (up to a maximum of 512 items).

##### 2.2.2.7.1.1.1.1.4.1 airsync:Class

The <airsync:Class> element<8> assigns the filters within the <airsync:Options> container to a given class.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <airsync:Options> (request only) | None | **String** | 0...1 (optional) |

Options for the same class within the same collection MUST NOT be redefined. The <airsync:Class> element is not necessary for the default items contained within the collection (contacts in a Contacts folder for example).

For example, to sync SMS messages, include class "SMS". To also sync e-mail messages at the same time, include another <airsync:Options> node with class "Email".

The valid <airsync:Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

##### 2.2.2.7.1.1.1.1.4.2 airsync:FilterType

The <airsync:FilterType> element specifies an optional time window in the **GetItemEstimate** command request for the objects sent from the server to the client.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <airsync:Options><9> | None | **Integer** | 0...1 (optional) |

The <airsync:FilterType> applies to e-mail, calendar, and task collections. If a filter type is specified, then the server sends an estimate of the items within the filter specifications.

If the <airsync:FilterType> element is present in the request, then the server manages objects on the client to maintain the specified time window. New objects are added to the client when they are within the time window. If the <airsync:FilterType> element is omitted, then all objects are sent from the server.

Calendar items that are in the future or that have recurrence, but no end date, are sent to the client regardless of the <airsync:FilterType> value.

The <airsync:FilterType> element does not support contact collections. However, if the <airsync:FilterType> element is included in a **GetItemEstimate** request for a contact collection, no error is returned.

The valid values for each collections type are listed in the following table.

| Value | Meaning | Applies to E-mail | Applies to calendar | Applies to tasks |
|---|---|---|---|---|
| 0 | No filter | Yes | Yes | Yes |
| 1 | 1 day ago | Yes | No, <Status> value 110 | No, <Status> value 110 |
| 2 | 3 days ago | Yes | No, <Status> value 110 | No, <Status> value 110 |
| 3 | 1 week ago | Yes | No, <Status> value 110 | No, <Status> value 110 |
| 4 | 2 weeks ago | Yes | Yes | No, <Status> value 110 |
| 5 | 1 month ago | Yes | Yes | No, <Status> value 110 |
| 6 | 3 months ago | No, <Status> value 110 | Yes | No, <Status> value 110 |
| 7 | 6 months ago | No, <Status> value 110 | Yes | No, <Status> value 110 |
| 8 | Incomplete tasks | No, <Status> value 110 | No, <Status> value 110 | Yes |

Specifying a <airsync:FilterType> of 9 or above for when the <CollectionId> identifies any e-mail, contact, calendar or task collection results in a <Status> value of 103.

#### 2.2.2.7.1.1.1.1.4.3 airsync:MaxItems

The <airsync:MaxItems> element<10> specifies the maximum number of items to include in the response. This element can only be included in a request when the <CollectionId> is set to *RI* to specify a recipient information **store**; otherwise, the server will respond with a status 2 error. The



Copyright © 2010 Microsoft Corporation.

value of <airsync:MaxItems> does not specify the limit of estimates available; rather, it only specifies the number of items, as a complete replacement would be double the number of items in the store (n deletes plus n additions).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <airsync:Options> | None | Integer | 0...1 (optional) |

Including <airsync:MaxItems> when the <CollectionId> is set to anything other than *RI* results in an invalid XML error, status code 2.

The result of including more than one <airsync:MaxItems> element as the child of the <airsync:Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.7.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **GetItemEstimate** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="GetItemEstimate:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="GetItemEstimate:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="GetItemEstimate">
          <xs:complexType>
               <xs:sequence>
                    <xs:element name="Response">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="Status" type="xs:unsignedByte"/>
                                   <xs:element name="Collection" minOccurs="0"
maxOccurs="unbounded">
                                        <xs:complexType>
                                             <xs:sequence>
                                                  <xs:element name="CollectionId" >
                                                    <xs:simpleType>
                                                      <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                      </xs:restriction>
                                                    </xs:simpleType>
                                                  </xs:element>
                                                  <xs:element name="Estimate"
type="xs:unsignedByte" />
                                             </xs:sequence>
                                        </xs:complexType>
                                   </xs:element>
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
</xs:schema>
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.7.2.1 GetItemEstimate

The <GetItemEstimate> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **GetItemEstimate** command.

##### 2.2.2.7.2.1.1 Response

The <Response> element wraps elements that describe estimated changes. Its child elements specify the status of the **GetItemEstimate** command request and information about the collection on which the estimate was made.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <GetItemEstimate> (response only) | <Status><br><Collection> | **Container** | 1 (required) |

###### 2.2.2.7.2.1.1.1 Status

The <Status> element indicates the success or failure of part or all of a **GetItemEstimate** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | None | **Unsigned byte** | 1 (required) |

If the entire request command fails, the <Status> element contains a value that indicates the type of global failure. However, if the failure occurs at the <Collection> level, a <Status> value is returned per <Collection>.

The following table lists the <Status> codes for the **GetItemEstimate** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | A collection was invalid or one of the specified collection IDs was invalid. | One or more of the specified folders does not exist or an incorrect folder was requested. | Item | Issue a **FolderSync** to get the new hierarchy. Then retry with a valid collection or collection ID. |
| 3 | The synchronization state has not been primed. | The client has issued a **GetItemEstimate** command without first issuing a **Sync** command request with a <airsync:SyncKey> value of zero (0). | Item | Issue a **Sync** command with synchronization key of zero (0) before issuing **GetItemEstimate** again. |
| 4 | The specified synchronization key was invalid. | Malformed or mismatched synchronization key.<br>—or— | Global | Issue a successful **Sync** command prior to issuing the **GetItemEstimate** command again. If the error is repeated, |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| | | The synchronization state is corrupted on the server. | | issue a **Sync** command with a <airsync:SyncKey> of zero (0). |

#### 2.2.2.7.2.1.1.2 Collection

The <Collection> element wraps elements that apply to a particular collection. A maximum of 300 <Collection> elements can be included in a single <Collections> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collections> (request only)<br><Response> (response only) | <airsync:SyncKey> (request only)<br><CollectionId><br><airsync:ConversationMode> (request only)<br><airsync:Options> (request only)<br><Estimate> (response only)<11> | **Container** | 0...300 (optional) |

##### 2.2.2.7.2.1.1.2.1 CollectionId

The <CollectionId> element specifies the server ID of the collection from which the item estimate is being obtained.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | **String** (Up to 64 characters) | 1…1 (required) |

The collection ID is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The <CollectionId> element is used in both **GetItemEstimate** command requests and responses.

##### 2.2.2.7.2.1.1.2.2 Estimate

The <Estimate> element specifies the estimated number of items in the collection or folder that have to be synchronized.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (response only) | None | **Unsigned byte** | 1 (required) |

### 2.2.2.8 ItemOperations

The **ItemOperations** command acts as a container for the <Fetch>, <EmptyFolderContents>, and <Move> elements to provide batched online handling of these operations against the server.

Operations that are contained within the **ItemOperations** element MUST be executed by the server in the specified order. The server MUST report the status per operation to the client. Accordingly, the client correlates these responses to the initial operation and proceeds appropriately.

The **ItemOperations** command cannot perform operations on items in the recipient information cache.



Copyright © 2010 Microsoft Corporation.

The <Fetch> operation is intended to be used on Microsoft Windows® SharePoint® Services technology or **Universal Naming Convention (UNC)** document metadata, search results, and items and attachments.

The <EmptyFolderContents> operation enables the client to empty a folder of all its items. Clients use <EmptyFolderContents> specifically to clear out all items in the Deleted Items folder if the user runs out of storage quota.

The <Move> operation moves a conversation to a destination folder.

The ItemOperations namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ItemOperations namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.8.1 Delivery of Content Requested by Fetch

Because the **ItemOperations** response potentially contains large amounts of binary data, the client can choose a delivery method that is most efficient for its implementation by providing the following two methods for delivering the content that is requested by the <Fetch> element:

- Inline
- Multipart

Inline

The inline method of delivering binary content is including base64-encoded data inside the WBXML. The inline approach generally requires the client to read the whole response into memory in order to parse it, thereby consuming a large amount of memory.

Multipart

The multipart method of delivering content is a multipart structure with the WBXML being the first part, and the requested data populating the subsequent parts. This format enables a client to handle large files without consuming large amounts of memory, because a file is read in pieces, one piece at a time.

The multipart approach enables the client to parse the small WBXML part, obtain references to the binary parts, and handle the binary parts as necessary, without reading the entire response into memory.

Multipart Request

If the client wants to have the document or documents returned in multipart format, the only difference between this request and the inline content request is the addition of the following HTTP header:

```
MS-ASAcceptMultiPart: T
```

If this header is not present, then the server uses the default of F (FALSE), and returns content inline. If the header is set to T (TRUE), then the server returns the document contents by using the multipart format.

The following is a sample request for the test.txt document in a UNC share:



Copyright © 2010 Microsoft Corporation.

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=administrator&DeviceId=v140Device&DeviceType=PocketPC
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0
MS-ASAcceptMultiPart: T
<ItemOperations>
 <Fetch>
  <Store>DocumentLibrary</Store>
  <documentlibrary:LinkId>\\feod31\public\test.txt</documentlibrary:LinkId>
 </Fetch>
</ItemOperations>
```

The following is a sample response to the request for the test.txt document.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.multipart

<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:documentlibrary="DocumentLibrary:" xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <documentlibrary:LinkId>\\contoso\test.txt</documentlibrary:LinkId>
      <Properties>
        <Range>0-999</Range>
        <Total>91646690</Total>
        <Part>1</Part>
        <Version>2009-11-09T09:00:00.692Z</Version>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

Multipart Response

At a high level, the multipart response consists of several key elements:

- HTTP headers that specify the content type (HTTP 'Content-Type' header) of the multipart response: application/vnd.ms-sync.multipart.
- Metadata consisting of a list of [integer, integer] tuples that specify the start and count of bytes, respectively, of each **body part**. The following is the format of the metadata:

```
'Number of Parts :' <number of body parts, including WBXML>
'Part' <part #> ':' <range>
```

Range specifies a [start, count] value that indicates the start and count of bytes for each body part. There MUST be at least one tuple, pointing to the WBXML response.

- The WBXML response, which contains status and application data for all requested items. The WBXML response is always the first part in the response. Items composed of binary content have a <Part> element that indicates the index (base 0) of the body part that corresponds to that item in the multipart structure. This index is used by the client to find the appropriate [start, count] entry in the metadata.



*Copyright © 2010 Microsoft Corporation.*

- Binary application data, which includes one or more binary data parts, the start and end byte of each of which is indicated in the WBXMLEX-Ranges header.

The following figure shows the elements of the multipart response.

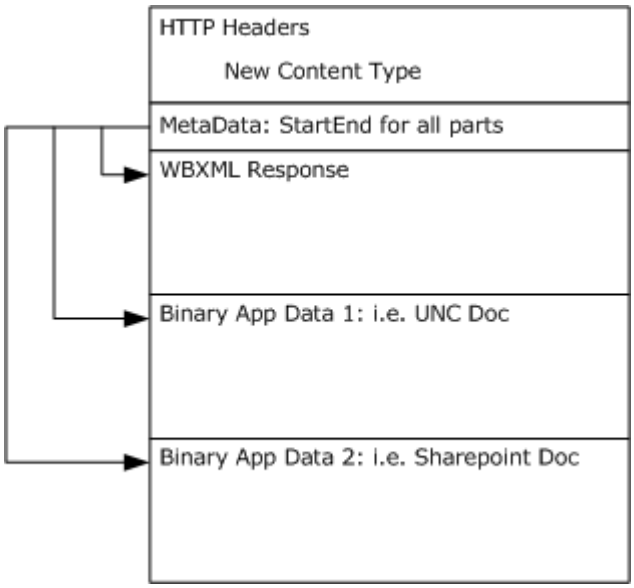


**Figure 1: ItemOperations command multipart response**

### 2.2.2.8.2 Request

The server that is implementing this protocol enforces the following XSD [XMLSCHEMA1] when it processes protocol requests.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        id="ItemOperations"
        targetNamespace="ItemOperations:"
        xmlns:search="Search:"
        xmlns:calendar="Calendar:"
        xmlns:email="Email:"
        xmlns:contacts2="Contacts2:"
        xmlns:contacts="Contacts:"
        xmlns:mstns="ItemOperations:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:documentlibrary="DocumentLibrary:"
        xmlns:airsync="AirSync:"
        xmlns:rm="RightsManagement:"
        xmlns:search="Search:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        attributeFormDefault="qualified"
        elementFormDefault="qualified">

    <xs:import namespace="DocumentLibrary:"/>
    <xs:import namespace="AirSync:"/>
    <xs:import namespace="AirSyncBase:"/>
    <xs:import namespace="Email:"/>
    <xs:import namespace="Calendar:"/>
```



Copyright © 2010 Microsoft Corporation.

```
    <xs:import namespace="Contacts:"/>
    <xs:import namespace="Contacts2:"/>
    <xs:import namespace="Search:"/>
    <xs:import namespace="RightsManagement:"/>

    <xs:element name="ItemOperations">
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="EmptyFolderContents">
                    <xs:complexType>
                        <xs:all>
                            <xs:element ref="airsync:CollectionId"/>
                            <xs:element name="Options" minOccurs="0">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element name="DeleteSubFolders"/>
                                    </xs:all>
                                </xs:complexType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Fetch" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="Store">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="1"/>
                                        <xs:maxLength value="256"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element ref="airsync:ServerId" minOccurs="0"/>
                            <xs:element ref="airsync:CollectionId" minOccurs="0"/>
                            <xs:element ref="documentlibrary:LinkId" minOccurs="0"/>
                            <xs:element ref="search:LongId" minOccurs="0"/>
                            <xs:element ref="airsyncbase:FileReference" minOccurs="0"/>
                            <xs:element name="Options" minOccurs="0">
                                <xs:complexType>
                                    <xs:choice maxOccurs="unbounded">
                                        <xs:element minOccurs="0" maxOccurs="unbounded"
name="Schema">
                                            <xs:complexType>
                                                <xs:choice maxOccurs="unbounded">
                                                    <xs:group
ref="email:TopLevelSchemaProps"/>
                                                    <xs:group
ref="airsyncbase:TopLevelSchemaProps"/>
                                                    <xs:group
ref="calendar:TopLevelSchemaProps"/>
                                                    <xs:group
ref="contacts:TopLevelSchemaProps"/>
                                                    <xs:group
ref="contacts2:TopLevelSchemaProps"/>
                                                </xs:choice>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="Range" minOccurs="0">
```



Copyright © 2010 Microsoft Corporation.

```
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:pattern value="[0-9]{1,9}-[0-
9]{1,9}"/>
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:element>
                                    <xs:element minOccurs="0" name="UserName">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:maxLength value="100" />
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:element>
                                    <xs:element minOccurs="0" name="Password">
                                        <xs:simpleType>
                                            <xs:restriction base="xs:string">
                                                <xs:maxLength value="256" />
                                            </xs:restriction>
                                        </xs:simpleType>
                                    </xs:element>
                                    <xs:element ref="airsync:MIMESupport" minOccurs="0"
/>
                                    <xs:element ref="airsyncbase:BodyPreference"
minOccurs="0" maxOccurs="256" />
                                    <xs:element ref="airsyncbase:BodyPartPreference"
minOccurs="0" />
                                    <xs:element ref="rm:RightsManagementSupport"
minOccurs="0"/>
                                </xs:choice>
                            </xs:complexType>
                        </xs:element>
                        <xs:element ref="rm:RemoveRightsManagementProtection"
minOccurs="0"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="Move">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="ConversationId" type="xs:string"/>
                        <xs:element name="DstFldId">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="64"/>
                                    <xs:minLength value="1"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="Options" minOccurs="0">
                            <xs:complexType>
                                <xs:all>
                                    <xs:element name="MoveAlways" minOccurs="0"/>
                                </xs:all>
                            </xs:complexType>
                        </xs:element>
                    </xs:all>
                </xs:complexType>
            </xs:element>
```



Copyright © 2010 Microsoft Corporation.

```
            </xs:choice>
          </xs:complexType>
        </xs:element>
    </xs:schema>
```

#### 2.2.2.8.2.1 ItemOperations

The <ItemOperations> element is the top-level element in the XML stream. The element identifies the body of the HTTP **POST** as containing an **ItemOperations** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <EmptyFolderContents> (request only)<br><Fetch> (request only)<br><Move> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1 (required) |

##### 2.2.2.8.2.1.1 EmptyFolderContents

The <EmptyFolderContents> element identifies the body of the request or response as containing the operation that deletes the contents of a folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request only)<br><Response> (response only) | <airsync:CollectionId><br><Options> (request only)<br><Status> (response only) | **Container** | 0...N (optional) |

The <EmptyFolderContents> response <Status> element uses the same values as the parent **ItemOperations** response <Status> element. For more details, see section 2.2.2.8.3.1.1.

The <EmptyFolderContents> element enables the client to empty a folder of all its items. The element supports a single <Options> element, <DeleteSubFolders>, which determines whether subfolders contained in the folder are deleted. If the <DeleteSubFolders> option is not included in the request, the subfolders of the specified <airsync:CollectionId> are not deleted.

Specifically, clients use <EmptyFolderContents> to empty the Deleted Items folder. The client can clear out all items in the Deleted Items folder when the user runs out of storage quota (indicated by the return of an HTTP 507 status code from the server).

###### 2.2.2.8.2.1.1.1 airsync:CollectionId

The <airsync:CollectionId> element enables a client to specify the folder to be emptied or the item to be fetched. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmptyFolderContents> | None | **String** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.2.1.1.2 Options

The <Options> element contains the options for the <EmptyFolderContents> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmptyFolderContents> (request only) | <DeleteSubFolders> (request only) | **Container** | 0...1 (optional) |

If the client specifies an option that is invalid for the parent element, the server returns a <Status> value of 2.

##### 2.2.2.8.2.1.1.2.1 DeleteSubFolders

The <DeleteSubFolders> element is a flag that indicates whether to delete the subfolders of the specified folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Flag** | 0...1 (optional) |

If the <DeleteSubFolders> element is not present in the request, the default behavior is to not delete subfolders.

#### 2.2.2.8.2.1.2 Fetch

The <Fetch> element retrieves an item from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request)<br><Response> (response) | <Store> (request only)<br><documentlibrary:LinkId> (optional)<br><search:LongId> (optional)<br><airsync:CollectionId> (optional)<br><airsync:ServerId> (optional)<br><Options> (request only)<br><Status> (response only)<br><airsync:Class> (response only)<br><Properties> (response only)<br><airsyncbase:FileReference> (request only)<br><rm:RemoveRightsManagementProtection> (request only) as specified in [MS-ASRM] section 2.2.2.3 | **Container** | 0...N (optional) |

The <Fetch> response <Status> element uses the same values as the parent **ItemOperations** response <Status> element. For more details, see section 2.2.2.8.3.1.1.

The <Fetch> operation is intended to be used on Microsoft Windows® SharePoint® Services technology or UNC document metadata, search results, and items and attachments.

Because the **ItemOperations** response potentially contains large amounts of binary data, this protocol enables the client to choose a delivery method that is most efficient for its implementation



Copyright © 2010 Microsoft Corporation.

by providing the following two methods to deliver content that is requested by the <Fetch> element:

- Inline—The binary content is base64 encoded and is included inside the WBXML.
- Multipart—This method involves a multipart structure in which the WBXML is the first part, and the requested data populates the subsequent parts. This format enables a client to handle large files without consuming large amounts of memory.

The inline approach generally requires the client to read the WBXML part into memory in order to parse it, thereby consuming a large amount of memory. The multipart approach enables the client to parse the small WBXML part, obtain references to the binary parts, and handle the binary parts as necessary, without reading the whole response into memory.

In the request, the client specifies the location and a byte range for the item. The location is indicated by either a link ID (<documentlibrary:LinkId> element) if the target item is identified by a **Uniform Resource Identifier (URI)**, or a file reference (<airsyncbase:FileReference> element) if the client is retrieving an e-mail attachment. The location is indicated by a server ID (<airsync:ServerId> element) if an ActiveSync ID is being used to identify the item.

The <Fetch> element supports several options, such as Byte ranges, Body preference, and Schema, as specified in section 2.2.2.8.2.1.2.7.

Multiple <Fetch> operations can be included within one **ItemOperations** request. In this case, the <Fetch> operations are executed in the order that is specified.

#### 2.2.2.8.2.1.2.1 Store

The <Store> element specifies the name of the store to which the parent operation applies.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (request only) | None | **String** (up to 256 characters) | 1...1 (required) |

The following values are valid for the <Store> element:

- Document Library (SharePoint and UNC links)
- Mailbox (items and attachments)

#### 2.2.2.8.2.1.2.2 airsync:ServerId

The <airsync:ServerId> element specifies a unique identifier that is assigned by the server to each object that can be synchronized or have an item operation applied to it. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **String** | 0...1 (optional) |

The client MUST store the <airsync:ServerId> for any item that is retrieved by means of the **Sync** command. In an **ItemOperations** request, the <airsync:ServerId> element can be used by the <Fetch> element to refer to the location of the item in question.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.2.1.2.3 airsync:CollectionId

The <airsync:CollectionId> element enables a client to specify the folder to be emptied or the item to be fetched. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (request only) | None | **String** | 0…1 (optional, as a child of <Fetch>) |

#### 2.2.2.8.2.1.2.4 documentlibrary:LinkId

The <documentlibrary:LinkId> element specifies a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or UNC documents.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **URI** | 0...1 (optional) |

The client MUST store the <documentlibrary:LinkId> that is retrieved by the **Search** command if the client will send requests using the <documentlibrary:LinkId> in the future. In an **ItemOperations** request, the <documentlibrary:LinkId> element can be used by the <Fetch> element to refer to the location of an item.

#### 2.2.2.8.2.1.2.5 search:LongId

The <search:LongId> element specifies a unique identifier that was assigned by the server to each result returned by a previous **Search** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **String** (up to 256 characters) | 0...1 (optional) |

#### 2.2.2.8.2.1.2.6 airsyncbase:FileReference

The <airsyncbase:FileReference> element specifies a unique identifier that is assigned by the server to each attachment to a given item.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (request only) | None | **String** | 0...1 (optional) |

The client MUST store the file reference for any item that is retrieved by means of the **Sync** or **Search** command. In an **ItemOperations** request, only one <airsyncbase:FileReference> identifier can exist per <Fetch> node. Violation of this constraint results in a <Status> value of 2 being returned from the server. The client can, however, retrieve multiple attachments by using one <Fetch> node per attachment.

If the <airsyncbase:FileReference> element is present, then <Range> is the only valid child element of <Options>.

#### 2.2.2.8.2.1.2.7 Options

The <Options> element contains the options for the <Fetch> operation.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (request only) | <Schema> (request only)<br><Range> (request only)<br><UserName> (request only)<br><Password> (request only)<br><airsync:MIMESupport> (request only)<br><airsyncbase:BodyPreference> (request only) as specified in [MS-ASAIRS] section 2.2.2.2<br><airsyncbase:BodyPartPreference> (request only) as specified in [MS-ASAIRS] section 2.2.2.3<br><rm:RightsManagementSupport> (request only) as specified in [MS-ASRM] section 2.2.2.1 | **Container** | 0...1 (optional) |

The following options are supported for <Fetch>:

- Schema
  - Per-class settings on format for <Fetch> results.
  - This protocol supports schemas for **Personal Information Manager (PIM)** items only; it does not support schemas for document library items or attachments.
  - Supports all top-level **property** nodes.

    For more information about the <Schema> element, see section 2.2.2.8.2.1.2.7.1.
- Byte ranges
  - Facilitates a checkpoint to improve the reliability of large data downloads.
  - This protocol supports ranges for document library items and attachments; it does not support ranges for other item types—that is, PIM items, such as e-mail, contact, calendar, or task items.
  - For attachments, the range applies to the file content.
  - For document library items, this applies to the file content.

    For more information about the <Range> element, see section 2.2.2.8.2.1.2.7.2.
- User name and password
  - Identifies the username and password required to fetch the desired item.
  - For more information about <UserName> and <Password> elements, see sections 2.2.2.8.2.1.2.7.3 and 2.2.2.8.2.1.2.7.4 respectively.
- MIME support
  - Indicates whether the server returns **MIME** content for S/MIME-only messages, all messages, or no messages
  - For more information about the <airsync:MIMESupport> element, see section 2.2.2.8.2.1.2.7.5.



Copyright © 2010 Microsoft Corporation.

- Body preference
  - Per-class settings on preferred body format.
  - This protocol supports body preferences for PIM items only; it does not support body preferences for document library items or attachments.
  - For more information about the <airsyncbase:BodyPreference> element, see [MS-ASAIRS] section 2.2.2.2.
- Body part preference
  - Per e-mail settings on preferred body part format.
  - When the <airsyncbase:BodyPartPreference> element is specified, the server returns the unique parts of the e-mail **message body** with reference to the parent item across all collections.
  - For more information about the <airsyncbase:BodyPartPreference> element, see [MS-ASAIRS] section 2.2.2.3.
- Rights management support
  - Indicates whether the device has information rights management enabled. If information rights management is enabled, the client requests that the information rights management content be returned in its schematized form in the response.

If the <airsyncbase:FileReference> element is present, then <Range> is the only valid child element of <Options>.

If the client specifies an option that is invalid for the parent element, the server returns a <Status> value of 2.

#### 2.2.2.8.2.1.2.7.1 Schema

The <Schema> element specifies the schema of the item to be fetched.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | Data elements from the content classes. For details about which of the elements from the content classes can be included, see [MS-ASCAL] section 3.1.5.1, [MS-ASCNTC] section 3.1.5.1, [MS-ASDOC] section 3.1.5.1, [MS-ASEMAIL] section 3.1.5.1, [MS-ASNOTE] section 3.1.5.1 and [MS-ASTASK] section 3.1.5.1. | **Container** | 0...N (optional) |

The <Schema> element is supported within options for PIM <Fetch> requests. It is not supported when the client is retrieving items from a document library or retrieving an attachment.

If <Schema> is not specified, the server allows all properties to be retrieved.

#### 2.2.2.8.2.1.2.7.2 Range

In an <ItemOperations> request, the <Range> element specifies the range of bytes that the client can receive in response to the <Fetch> operation for a document library item. In an



Copyright © 2010 Microsoft Corporation.

<ItemOperations> response, the <Range> element specifies the actual range of bytes for an item that is contained in a given <Fetch> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Properties> (response only) | None | String in the format m-n, where m<n, and m is the minimum value and n is the maximum value. | 0...1 (optional) |

The server provides a best effort at fulfilling the request. Therefore, the client cannot assume that the byte-range that is specified in the request exactly matches the byte-range that is returned in the response. The byte-range that is specified by the server in the response is the authoritative value.

If <Range> is omitted in the <Fetch> request, the whole item is fetched.

If the <airsyncbase:FileReference> element is present, then <Range> is the only valid child element of <Options>.

#### 2.2.2.8.2.1.2.7.3 UserName

The <UserName> element specifies the username of the account leveraged to fetch the desired item. The <Password> element contains the corresponding account password.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **String** (up to 100 characters) | 0...1 (optional) |

#### 2.2.2.8.2.1.2.7.4 Password

The <Password> element specifies the password for the given <UserName>.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **String** (up to 256 characters) | 0...1 (optional) |

The server accepts password values up to 256 characters in length. However, logon dialog boxes can limit password lengths to a smaller value.

#### 2.2.2.8.2.1.2.7.5 airsync:MIMESupport

The <airsync:MIMESupport> element is included in the <Options> element of a client **ItemOperations** command request to enable MIME support for e-mail items that are sent from the server to the client. For an example, see section 4.10.2.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Unsigned Byte** | 0...1 (optional) |

The following table lists the valid values for this element.



Copyright © 2010 Microsoft Corporation.

| Value | Meaning |
|---|---|
| 0 | Never send MIME data. |
| 1 | Send MIME data for S/MIME messages only. Send regular body for all other messages. |
| 2 | Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution. |

To support fetching of the full S/MIME message, the <Fetch> request MUST include the following elements in the <Options> element:

- The <airsync:MIMESupport> element to indicate to the server to return MIME for S/MIME-only messages, all messages, or no messages.
- The <airsyncbase:BodyPreference> element with its child element, <Type> having a value of 4 to inform the server that the device can read the MIME **binary large object (BLOB)**.

The server's response MUST include the <airsyncbase:Body> element, which is a child of the <Properties> element. The <airsyncbase:Body> element is a complex element and MUST contain the following child nodes in an S/MIME <Fetch> response:

- The <airsyncbase:Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <airsyncbase:EstimatedDataSize> element to specify the rough total size of the data.
- The <airsyncbase:Data> element that contains the full MIME BLOB.

For more details about the <airsyncbase:Body> element or the <airsyncbase:BodyPreference> element, see [MS-ASAIRS] section 2.2.2.4 or 2.2.2.2, respectively.

#### 2.2.2.8.2.1.3 Move

The <Move> element identifies the body of the request or response as containing the operation that moves a given conversation.<12>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request only)<br><Response> (response only) | <ConversationId><br><DstFldId> (request only)<br><Options> (request only)<br><Status> (response only) | **Container** | 0...N (optional) |

##### 2.2.2.8.2.1.3.1 ConversationId

The <ConversationId> element specifies the conversation to be moved.<13>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request or response) | None | **String** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.2.1.3.2 DstFldId

The <DstFldId> element specifies the destination folder where the conversation is to be moved.<14>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only) | None | **String** (up to 64 characters) | 1...1 (required) |

#### 2.2.2.8.2.1.3.3 Options

The <Options> element contains the options for the <Move> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only) | <MoveAlways> (request only) | **Container** | 0...1 (optional) |

If the client specifies an option that is invalid for the parent element, the server returns a <Status> value of 2.

##### 2.2.2.8.2.1.3.3.1 MoveAlways

The <MoveAlways> element is a flag that indicates whether to always move the specified conversation, including all future e-mails in the conversation, to the <DstFldId> folder.<15>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Empty tag** | 0...1 (optional) |

The <MoveAlways> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <MoveAlways> tag.

The <MoveAlways> element MUST be included in an **ItemOperations** request when performing a move operation on a conversation. A <Status> value of 155 is returned if the <MoveAlways> element is not included in the **ItemOperations** request for a move operation.

### 2.2.2.8.3 Response

The following code shows the XSD [XMLSCHEMA1] for the **ItemOperations** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ItemOperations:"
attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="ItemOperations:" xmlns:airsync="AirSync:" xmlns:airsyncbase="AirSyncBase:"
xmlns:documentlibrary="DocumentLibrary:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="DocumentLibrary:" />
  <xs:import namespace="AirSync:" />
  <xs:import namespace="AirSyncBase:" />
  <xs:element name="ItemOperations">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:integer "/>
        <xs:element name="Response" minOccurs="0">
          <xs:complexType>
```



Copyright © 2010 Microsoft Corporation.

```
            <xs:sequence>
              <xs:element name="Move" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Status" type="xs:integer"/>
                    <xs:element name="ConversationId" type="xs:string"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="EmptyFolderContents" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Status" type="xs:integer"/>
                    <xs:element ref="airsync:CollectionId"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
               <xs:element name="Fetch" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Status" type="xs:integer"/>
                    <xs:element ref="airsync:CollectionId"/>
                    <xs:element ref="airsync:ServerId" minOccurs="0"/>
                    <xs:element ref="airsync:Class" minOccurs="0"/>
                    <xs:element ref="documentlibrary:LinkId" minOccurs="0"/>
                    <xs:element name="Properties" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Range" type="xs:string" minOccurs="0"/>
                          <xs:element name="Total" type="xs:integer" minOccurs="0"/>
                          <xs:element name="Data" type="xs:string" minOccurs="0"/>
                          <xs:element name="Part" type="xs:integer" minOccurs="0"/>
                          <xs:element name="Version" type="xs:dateTime" minOccurs="0"/>
                          <xs:element ref="airsyncbase:Body" minOccurs="0"/>
                          <xs:element ref="airsyncbase:BodyPart" minOccurs="0"/>
                          <!-- Data elements are from the content classes and can be included
as children of the Properties element. For details about the content classes, see [MS-ASCAL],
[MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]-->
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
 </xs:schema>
```

#### 2.2.2.8.3.1 ItemOperations

The <ItemOperations> element is the top-level element in the XML stream. The element identifies the body of the HTTP **POST** response as containing an **ItemOperations** command.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <EmptyFolderContents> (request only)<br><Fetch> (request only)<br><Move> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1...1 (required) |

#### 2.2.2.8.3.1.1 Status

The <Status> element contains a code that indicates the success or failure of the **ItemOperations** command and the operations within the **ItemOperations** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (response only) | None | **Integer** | 0...1 |

The following table lists the <Status> codes for the **ItemOperations** command. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Status Code | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error - protocol violation/XML validation error. |
| 3 | Server error. |
| 4 | Document library access - The specified URI is bad. |
| 5 | Document library - Access denied. |
| 6 | Document library - The object was not found or access denied. |
| 7 | Document library - Failed to connect to the server. |
| 8 | Document library - The byte-range is invalid or too large. |
| 9 | Document library - The store is unknown or unsupported. |
| 10 | Document library - The file is empty. |
| 11 | Document library - The requested data size is too large. |
| 12 | Document library - Failed to download file because of input/output (I/O) failure. |
| 14 | Mailbox fetch provider - The item failed conversion. |
| 15 | Attachment fetch provider - Attachment or attachment ID is invalid. |
| 16 | Policy-related - Server blocked access. |
| 17 | Empty folder contents - Partial success; the command completed partially. |
| 18 | Credentials required. |



Copyright © 2010 Microsoft Corporation.

| Status Code | Meaning |
|---|---|
| 155 | Protocol error. The <Options> element and the <MoveAlways> element are missing from the **ItemOperations** request. |
| 156 | Action not supported. The destination folder MUST be of type IPF.Note. For more information about folder types, see [MS-OXOSFLD] section 2.2.5. |

#### 2.2.2.8.3.1.2 Response

The <Response> element is a container for the operation responses.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (response only) | <EmptyFolderContents><br><Fetch><br><Move> | **Container** | 0...1 (optional) |

##### 2.2.2.8.3.1.2.1 Move

The <Move> element identifies the body of the request or response as containing the operation that moves a conversation to a folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request only)<br><Response> (response only) | <ConversationId><br><Options> (request only)<br><Status> (response only) | **Container** | 0...N (optional) |

###### 2.2.2.8.3.1.2.1.1 Status

The <Status> element contains a code that indicates the success or failure of the <Move> operation within the **ItemOperations** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (response only) | None | **Integer** | 0...1 |

The following table lists the <Status> codes for the **ItemOperations** command. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Status Code | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error - protocol violation/XML validation error. |
| 3 | Server error. |
| 4 | Document library access - The specified URI is bad. |



Copyright © 2010 Microsoft Corporation.

| Status Code | Meaning |
|---|---|
| 5 | Document library - Access denied. |
| 6 | Document library - The object was not found or access denied. |
| 7 | Document library - Failed to connect to the server. |
| 8 | Document library - The byte-range is invalid or too large. |
| 9 | Document library - The store is unknown or unsupported. |
| 10 | Document library - The file is empty. |
| 11 | Document library - The requested data size is too large. |
| 12 | Document library - Failed to download file because of input/output (I/O) failure. |
| 14 | Mailbox fetch provider - The item failed conversion. |
| 15 | Attachment fetch provider - Attachment or attachment ID is invalid. |
| 16 | Policy-related - Server blocked access. |
| 17 | Empty folder contents - Partial success; the command completed partially. |
| 18 | Credentials required. |
| 155 | Protocol error. The <Options> element and the <MoveAlways> element are missing from the **ItemOperations** request. |
| 156 | Action not supported. The destination folder MUST be of type IPF.Note. For more information about folder types, see [MS-OXOSFLD] section 2.2.5. |

#### 2.2.2.8.3.1.2.1.2 ConversationId

The <ConversationId> element specifies the conversation to be moved.<16>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request or response) | None | **String** | 1…1 (required) |

#### 2.2.2.8.3.1.2.2 EmptyFolderContents

The <EmptyFolderContents> element identifies the body of the request or response as containing the operation that deletes the contents of a folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request only)<br><Response> (response only) | <airsync:CollectionId><br><Options> (request only)<br><Status> (response only) | <Container> | 0...N (optional) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.3.1.2.2.1 Status

The <Status> element contains a code that indicates the success or failure of the <EmptyFolderContents> operation within the **ItemOperations** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmptyFolderContents> (response only) | None | **Integer** | 0...1 |

The following table lists the <Status> codes for the **ItemOperations** command. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Status Code | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error - protocol violation/XML validation error. |
| 3 | Server error. |
| 4 | Document library access - The specified URI is bad. |
| 5 | Document library - Access denied. |
| 6 | Document library - The object was not found or access denied. |
| 7 | Document library - Failed to connect to the server. |
| 8 | Document library - The byte-range is invalid or too large. |
| 9 | Document library - The store is unknown or unsupported. |
| 10 | Document library - The file is empty. |
| 11 | Document library - The requested data size is too large. |
| 12 | Document library - Failed to download file because of input/output (I/O) failure. |
| 14 | Mailbox fetch provider - The item failed conversion. |
| 15 | Attachment fetch provider - Attachment or attachment ID is invalid. |
| 16 | Policy-related - Server blocked access. |
| 17 | Empty folder contents - Partial success; the command completed partially. |
| 18 | Credentials required. |
| 155 | Protocol error. The <Options> element and the <MoveAlways> element are missing from the **ItemOperations** request. |
| 156 | Action not supported. The destination folder MUST be of type IPF.Note. For more information about folder types, see [MS-OXOSFLD] section 2.2.5. |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.3.1.2.2.2 airync:CollectionId

The <airsync:CollectionId> element identifies the folder that was emptied. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmptyFolderContents> | None | **String** | 1 (required) |

#### 2.2.2.8.3.1.2.3 Fetch

The <Fetch> element retrieves an item from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ItemOperations> (request)<br><Response> (response) | <Store> (request only)<br><documentlibrary:LinkId> (optional)<br><search:LongId> (optional)<br><airsync:CollectionId> (optional)<br><airsync:ServerId> (optional)<br><Options> (request only)<br><Status> (response only)<br><airsync:Class> (response only)<br><Properties> (response only)<br><airsyncbase:FileReference> (request only)<br><rm:RemoveRightsManagementProtection> (request only) as specified in [MS-ASRM] section 2.2.2.3 | **Container** | 0...N (optional) |

The <Fetch> element supports several options, such as Byte ranges, Body preference, and Schema as specified in section 2.2.2.8.2.1.2.7.

The <Fetch> response contains either the requested byte range of the item, or an error code that indicates why the fetch failed.

For more information about the <Fetch> element, see sections 2.2.2.8.1 and 2.2.2.8.2.1.2.

#### 2.2.2.8.3.1.2.3.1 Status

The <Status> element contains a code that indicates the success or failure of the <Fetch> operation within the **ItemOperations** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (response only) | None | **Integer** | 0...1 |

The following table lists the <Status> codes for the **ItemOperations** command. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.



Copyright © 2010 Microsoft Corporation.

| Status Code | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error - protocol violation/XML validation error. |
| 3 | Server error. |
| 4 | Document library access - The specified URI is bad. |
| 5 | Document library - Access denied. |
| 6 | Document library - The object was not found or access denied. |
| 7 | Document library - Failed to connect to the server. |
| 8 | Document library - The byte-range is invalid or too large. |
| 9 | Document library - The store is unknown or unsupported. |
| 10 | Document library - The file is empty. |
| 11 | Document library - The requested data size is too large. |
| 12 | Document library - Failed to download file because of input/output (I/O) failure. |
| 14 | Mailbox fetch provider - The item failed conversion. |
| 15 | Attachment fetch provider - Attachment or attachment ID is invalid. |
| 16 | Policy-related - Server blocked access. |
| 17 | Empty folder contents - Partial success; the command completed partially. |
| 18 | Credentials required. |
| 155 | Protocol error. The <Options> element and the <MoveAlways> element are missing from the **ItemOperations** request. |
| 156 | Action not supported. The destination folder MUST be of type IPF.Note. For more information about folder types, see [MS-OXOSFLD] section 2.2.5. |

#### 2.2.2.8.3.1.2.3.2 airsync:Collectionid

The <airsync:CollectionId> element identifies the folder that was fetched. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (request only) | None | **String** | 0…1 (optional) |

#### 2.2.2.8.3.1.2.3.3 airsync:ServerId

The <airsync:ServerId> element specifies a unique identifier that is assigned by the server to each object that can be synchronized or have an item operation applied to it. This element is defined in the AirSync namespace.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **String** | 0...1 (optional) |

The client MUST store the <airsync:ServerId> for any item that is retrieved by means of the **Sync** or **Search** command. In an **ItemOperations** request, the <airsync:ServerId> element can be used by the <Fetch> element to refer to the location of the item in question.

#### 2.2.2.8.3.1.2.3.4 airsync:Class

In a response, the <airsync:Class> element indicates the class of the content of the fetched item. This element is defined in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (response only) | None | **String** | 0...1 (optional) |

The following are valid values for the <airsync:Class> element in a response.

- *E-mail*
- *Contacts*
- *Calendar*
- *Tasks*

#### 2.2.2.8.3.1.2.3.5 documentlibrary:LinkId

The <documentlibrary:LinkId> element specifies a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or UNC documents.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **URI** | 0...1 (optional) |

The client MUST store the <documentlibrary:LinkID> that is retrieved by the **Search** command if the client will send requests by using the <documentlibrary:LinkID> in the future. In an **ItemOperations** request, the <documentlibrary:LinkId> element can be used by the <Fetch> element to refer to the location of an item.

#### 2.2.2.8.3.1.2.3.6 Properties

The <Properties> element contains a list of the schema properties for the item that the client wants to have returned in the <Fetch> response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> (response only) | The schema properties of the item being fetched (request only)<br><Range> (response only)<br><Data> (response only)<br><Part> (response only) | **Container** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <Version> (response only)<br><Total> (response only)<br><airsyncbase:Body> (response only)<br><airsyncbase:BodyPart> (response only)<br>Data elements are from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]. | | |

#### 2.2.2.8.3.1.2.3.6.1 Range

In an **ItemOperations** request, the <Range> element specifies the range of bytes that the client can receive in response to the <Fetch> operation for a document library item. In an **ItemOperations** response, the <Range> element specifies the actual range of bytes for an item that is contained in a given <Fetch> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request)<br><Properties> (response) | None | **String** in the format m-n, where m<n, and m is the minimum value and n is the maximum value. The byte range is zero-indexed; the first byte is indicated by a 0 (zero). | 0...1 (optional) |

The server provides a best effort at fulfilling the request. Therefore, the client cannot assume that the byte-range specified in the request will exactly match the byte-range returned in the response. The byte-range that is specified by the server in the response is the authoritative value.

If <Range> is omitted in the <Fetch> request, the entire item is fetched.

#### 2.2.2.8.3.1.2.3.6.2 Total

The <Total> element indicates the total size of an item on the server, in bytes.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Properties> (response only) | None | **Integer** | 0...1 (optional) |

#### 2.2.2.8.3.1.2.3.6.3 Data

The <Data> element contains the item content for inline content responses.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Properties> (response only) | None | **String** | 0...1 (optional) |

The content of the <Data> element is a base64 encoding of the binary document, attachment, or body data. The size of the data (in bytes) that is contained within the <Data> element is indicated by the <Range> element in the fetch response. The total size of the item (in bytes) is indicated by the <Total> element.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.8.3.1.2.3.6.4 Part

The <Part> element specifies an integer index into the metadata of the multipart response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Properties> (response only) | None | **Integer** | 0...1 (optional) |

The <Part> element is present only in a multipart **ItemOperations** response.

The <Part> element can be used to locate the [start, end] tuple that specifies the starting byte and ending byte for this item's binary content in the command response.

#### 2.2.2.8.3.1.2.3.6.5 Version

The <Version> element is a **dateTime** stamp that indicates the time at which a document item was last modified.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Properties> (response only) | None | **DateTime** | 0...1 (optional) |

The <Version> element is present only in the response and only when **ItemOperations** is used to access a Windows SharePoint Services or UNC resource.

### 2.2.2.9 MeetingResponse

The **MeetingResponse** command is used to accept, tentatively accept, or decline a meeting request in the user's Inbox folder or Calendar folder.<17>

The **MeetingResponse** command can only be used when the <CollectionId> element is being used to synchronize the folder that contains the meeting request item.

The **SendMail** command can be used to send a message back to the meeting **organizer**, notifying him or her that the meeting request was accepted or declined.

The MeetingResponse namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the MeetingResponse namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.9.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **MeetingResponse** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="MeetingResponse:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="MeetingResponse:"
        xmlns:search="Search:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:import namespace="Search:"/>

    <xs:element name="MeetingResponse">
```



Copyright © 2010 Microsoft Corporation.

```
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Request">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="UserResponse">
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedByte">
                                        <xs:enumeration value="3"/>
                                        <xs:enumeration value="1"/>
                                        <xs:enumeration value="2"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element minOccurs="0" name="CollectionId">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element minOccurs="0" name="RequestId">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element ref="search:LongId" minOccurs="0"/>
                            <xs:element name="InstanceId" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="24"/>
                                        <xs:maxLength value="24"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.9.1.1 MeetingResponse

The <MeetingResponse> element is the top-level element in the XML stream. The element identifies the body of the HTTP **POST** as containing a **MeetingResponse** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Request> (request only) | **Container** | 1…1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.9.1.1.1 Request

The <Request> element is a container for elements in a **MeetingResponse** command request. Its child elements specify the meeting request that is being responded to, the response to that meeting request, and the folder on the server that the meeting request is located in.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <MeetingResponse> (request only) | <UserResponse> (request only)<br><CollectionId> (request only)<br><RequestId> (request only)<br><search:LongId> (request only) as specified in section 2.2.2.14.2.1.2.1.2.2<br><InstanceId> (request only) | **Container** | 1...n (required) |

#### 2.2.2.9.1.1.1.1 UserResponse

The <UserResponse> element indicates in the **MeetingResponse** command request whether the meeting is being accepted, tentatively accepted, or declined.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only) | None | **Integer** | 1 (required) |

The following table shows valid values for the element.

| Value | Meaning |
|---|---|
| 1 | Accepted |
| 2 | Tentatively accepted |
| 3 | Declined |

#### 2.2.2.9.1.1.1.2 CollectionId

The <CollectionId> element specifies the folder that contains the meeting request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only) | None | **String** (Up to 64 characters) | 0...1 (Required, or optional if <search:LongId> is included, as specified in section 2.2.2.14.2.1.2.1.2.2) |

Because meeting requests are most commonly sent to the Inbox folder, the <CollectionId> value that specifies the Inbox folder is the most common value for this element.

The <CollectionId> is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.9.1.1.1.3 RequestId

The <RequestId> element specifies the server ID of the meeting request message item.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only)<br><Result> (response only) | None | **String** (Up to 64 characters) | 0...1 (Required, or optional if <search:LongId> is included, as specified in section 2.2.2.14.2.1.2.1.2.2) |

When the client sends a **MeetingResponse** command request, the client includes a <RequestId> element to identify which meeting request is being responded to. The <RequestId> element is also returned in the response to the client along with the status of the user's response to the meeting request.

#### 2.2.2.9.1.1.1.4 InstanceId

The <InstanceId> element<18> specifies the instance of the recurring meeting to be modified.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only) | None | **String** | 0...1 (optional) |

The <InstanceId> element contains the start time of the **appointment** or meeting instance to be modified. If the <InstanceId> element is not included in the **MeetingResponse** request, then the action is to be taken on every instance of the recurring item. The <InstanceId> element can specify the start time of an exception to a recurring appointment or meeting. The <InstanceId> element can be used with the <search:LongId> element to identify a calendar item, or it can be used with the <CollectionId> and <RequestId> elements to identify a calendar item.

The format of <InstanceId> is a **string** in **dateTime** format with the punctuation separators, for example, 2010-04-08T18:16:00.000Z. If the <InstanceId> value specified is not in the proper format, the server responds with a <Status> value of 104. If the <InstanceId> value specifies a non-recurring meeting, the server responds with a <Status> value of 146.

### 2.2.2.9.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **MeetingResponse** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="MeetingResponse:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
     targetNamespace="MeetingResponse:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="MeetingResponse">
          <xs:complexType>
               <xs:sequence>
                    <xs:element name="Result" maxOccurs="unbounded">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="RequestId">
                                      <xs:simpleType>
                                         <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:element>
                              <xs:element name="Status" type="xs:unsignedByte"/>
                              <xs:element name="CalendarId" minOccurs="0">
                                 <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                       <xs:maxLength value="64"/>
                                    </xs:restriction>
                                 </xs:simpleType>
                              </xs:element>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:schema>
```

#### 2.2.2.9.2.1 MeetingResponse

The <MeetingResponse> element is the top-level element in the XML stream. The element identifies the body of the HTTP **POST** as containing a **MeetingResponse** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Result> (response only) | **Container** | 1…1 (required) |

#### 2.2.2.9.2.1.1 Result

The <Result> element is a container for elements that are sent to the client in a **MeetingResponse** command response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <MeetingResponse> (response only) | <RequestId> (response only)<br><Status> (response only)<br><CalendarId> (response only) | **Container** | 1...N (required) |

The <Result> element's child elements identify the meeting request message item on the server and the status of the response to the meeting request. If the meeting request is accepted, the server ID of the calendar item is also returned.

#### 2.2.2.9.2.1.1.1 RequestId

The <RequestId> element specifies the server ID of the meeting request message item.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Request> (request only)<br><Result> (response only) | None | **String** (Up to 64 characters) | 1 (required) |



Copyright © 2010 Microsoft Corporation.

When the client sends a **MeetingResponse** command request, the client includes a <RequestId> element to identify which meeting request is being responded to. The <RequestId> element is also returned in the response to the client along with the status of the user's response to the meeting request.

#### 2.2.2.9.2.1.1.2 Status

The <Status> element indicates the success or failure of the **MeetingResponse** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Result> (response only) | None | **Integer** | 1...N (required) |

The following table lists the <Status> value codes for the **MeetingResponse** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | Invalid meeting request. | The client has sent a malformed or invalid item.<br>The request is referencing an item other than a meeting request, e-mail, or calendar item.<br>The request points to an appointment in which the user is the organizer.<br>The <InstanceId> element specifies an e-mail meeting request item.<br>The <InstanceId> element specifies a nonexistent instance or is null. | Item | Stop sending the item. This is not a transient condition. |
| 3 | An error occurred on the server mailbox. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **MeetingResponse**. If continued attempts fail, synchronize the folder again, and then attempt the MeetingResponse command again. If it still continues to fail, make no changes. |
| 4 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the **MeetingResponse**. If continued attempts fail, synchronize the folder again, and then attempt the MeetingResponse command again. If it still continues to fail, make no changes. |

#### 2.2.2.9.2.1.1.3 CalendarId

The <CalendarId> element specifies the server ID of the calendar item.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Result> (response only) | None | **String** (Up to 64 characters) | 0...1 |

The following table shows valid values for the element.

| Value | Meaning |
|---|---|
| 1 | Success |

The <CalendarId> element is included in the **MeetingResponse** command response that is sent to the client if the meeting request was not declined. If the meeting is accepted or tentatively accepted, the server adds a new item to the calendar and returns its server ID in the <CalendarId> element in the response. If the client created a tentative meeting calendar item, the client updates that item with the new server ID. The client also changes the busy status from tentative to busy. When a meeting is accepted, the server also creates a new calendar item with the same server ID. This means there is a conflict that will be resolved the next time the calendar is synchronized.

If the meeting is declined, the response does not contain a <CalendarId>.

### 2.2.2.10 MoveItems

The **MoveItems** command moves an item or items from one folder on the server to another.

The item to be moved is identified by its server ID in the **MoveItems** command request. The source and destination folders are also identified by their server IDs in the command request. The **MoveItems** command response shows the status of the move, the message that was moved, and the new **message ID**.

When items are moved between folders on the server, the client receives <Delete> and <Add> operations the next time the client synchronizes the affected folders.

An item that has been successfully moved to a different folder can be assigned a new server ID by the server.

The Move namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the Move namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.10.1 Request

The following code shows the XSD for the **MoveItems** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Move:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Move:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="MoveItems">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Move">
                    <xs:complexType>
                        <xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
                                <xs:element name="SrcMsgId">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                            <xs:minLength value="1"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="SrcFldId">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                            <xs:minLength value="1"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                                <xs:element name="DstFldId">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                            <xs:minLength value="1"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

#### 2.2.2.10.1.1 MoveItems

The <MoveItems> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **MoveItems** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Move> (request only)<br><Response> (response only) | **Container** | 1 (required) |

#### 2.2.2.10.1.1.1 Move

The <Move> element is a container for elements that describe details of the items to be moved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <MoveItems> (request only) | <SrcMsgId> (request only)<br><SrcFldId> (request only)<br><DstFldId> (request only) | **Container** | 1...N (required) |

The <Move> element's child elements specify the item to be moved, the folder it's currently located in, and the folder it will be moved to.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.10.1.1.1.1 SrcMsgId

The <SrcMsgId> element specifies the server ID of the item to be moved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only)<br><Response> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

#### 2.2.2.10.1.1.1.2 SrcFldId

The <SrcFldId> element specifies the server ID of the source folder (that is, the folder that contains the items to be moved).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The server ID of the source folder is obtained from the <folderhierarchy:ServerId> element of a previous **FolderSync** or **FolderCreate** command.

#### 2.2.2.10.1.1.1.3 DstFldId

The <DstFldId> element specifies the server ID of the destination folder (that is, the folder to which the items are moved).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The server ID of the destination folder is obtained from the <folderhierarchy:ServerId> element of a previous **FolderSync** or **FolderCreate** command.

### 2.2.2.10.2 Response

The following code shows the XSD for the **MoveItems** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="Move:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="Move:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="MoveItems">
          <xs:complexType>
               <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="Response">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="SrcMsgId">
                                      <xs:simpleType>
                                         <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                            <xs:minLength value="1"/>
                                         </xs:restriction>
                                      </xs:simpleType>
                                   </xs:element>
```



Copyright © 2010 Microsoft Corporation.

```
                                    <xs:element name="Status" type="xs:unsignedByte" />
                                    <xs:element name="DstMsgId">
                                       <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                              <xs:maxLength value="64"/>
                                              <xs:minLength value="1"/>
                                           </xs:restriction>
                                        </xs:simpleType>
                                     </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:schema>
```

#### 2.2.2.10.2.1 MoveItems

The <MoveItems> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **MoveItems** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Move> (request only)<br><Response> (response only) | **Container** | 1 (required) |

##### 2.2.2.10.2.1.1 Response

The <Response> element serves as a container for elements that describe the moved items.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <MoveItems> (response only) | <SrcMsgId> (response only)<br><Status> (response only)<br><DstMsgId> (response only) | **Container** | 1 (required) |

###### 2.2.2.10.2.1.1.1 SrcMsgId

The <SrcMsgId> element specifies the server ID of the item that was moved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Move> (request only)<br><Response> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

###### 2.2.2.10.2.1.1.2 Status

The <Status> element indicates the success or failure of an item moved. If the command failed, <Status> contains a code indicating the type of failure.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | None | **Unsigned byte** | 1 (required) |

The following table lists the <Status> codes for the **MoveItems** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Invalid source collection ID. | The source folder <CollectionId> is not recognized by the server, possibly because the source folder has been deleted. | Item | Issue a **FolderSync** command to get the new hierarchy. Then, sync the folder and reissue the **MoveItems** request if the items are still present in this source collection. |
| 2 | Invalid destination collection ID. | The destination folder <CollectionId> is not recognized by the server, possibly because the source folder has been deleted. | Item | Issue a **FolderSync** to get the new hierarchy. Then, use a valid collectionID. |
| 3 | Success. | Server successfully completed command. | Global | None. |
| 4 | Source and destination collection IDs are the same. | The client supplied a destination folder that is the same as the source. | Item | Only send requests where the <CollectionId> for the source and destination differ. |
| 5 | One of the following failures occurred: the item cannot be moved to more than one item at a time, or the source or destination item was locked. | More than one <DstFldId> was included in the request or an item with that name already exists. | Global | Retry the **MoveItems** request with only one <DstFldId> element or move the item to another location. |
| 7 | Source or destination item was locked. | Transient server condition. | Item | Retry. |

#### 2.2.2.10.2.1.1.3 DstMsgId

The <DstMsgId> element specifies the new server ID of the item after the item is moved to the destination folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | None | **String** (Up to 64 characters) | 0…1 (optional) |

### 2.2.2.11 Ping

The **Ping** command is used to request that the server monitor specified folders for changes that would require the client to resynchronize.



Copyright © 2010 Microsoft Corporation.

The body of the request contains a list of folders on the server about which the client is requesting notifications and an interval of time that specifies how long the server SHOULD wait before responding.

The server does not immediately issue a response to the client's **Ping** request. Instead, the server waits until one of two events occur: either the time-out that is specified by the client elapses, or changes occur in one of the folders that the client specifies. The response that the server issues indicates which of these events has happened so that the client can react appropriately.

The server uses the last <airsync:SyncKey> returned to the client when determining to report in the **Ping** response. Therefore the client MUST have received the response to its last **Sync** request and successfully applied the changes sent by the server, prior to issuing a **Ping** request.

In the case of no changes on the server, the client can then reissue a new **Ping** request. In the case of changes, the response indicates in which folders those changes occurred so that the client can resynchronize those folders.

Note that if no changes occur in any of the folders that are specified by the client for a significant length of time (longer than the value of the <HeartbeatInterval> element), the client runs in a loop in which it issues a **Ping** request, receives a response indicating that there are no changes, and then reissues the **Ping** request. This loop is called the heartbeat. The length of time that the server waits before issuing a response is called the heartbeat interval. For more details about the <HeartbeatInterval> element, see section 2.2.2.11.1.1.1.

To reduce the amount of data sent in a **Ping** command request, the server caches the heartbeat interval and folder list. The client can omit the heartbeat interval, the folder list, or both from subsequent **Ping** requests if those parameters have not changed from the previous **Ping** request. If neither the heartbeat interval nor the folder list has changed, the client can issue an empty **Ping** request – one with no XML body. The server will use the previously cached XML sent by the client if it receives an empty **Ping** request.

If the **Ping** element is specified in an XML request body, either the <HeartbeatInterval> element or the <Folders> element or both MUST be specified.

The Ping namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the Ping namespace use the namespace prefixes defined in section 2.2.1.

### 2.2.2.11.1 Request

A **Ping** command can be sent with no body, in which case the cached version is used. This XSD [XMLSCHEMA1] is applied only to requests with a body.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Ping:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Ping:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="Ping">
        <xs:complexType>
            <xs:all>
                <xs:element name="HeartbeatInterval" minOccurs="0">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer"/>
                    </xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
                </xs:element>
                <xs:element name="Folders" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Folder" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element name="Id">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="64"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="Class">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:enumeration value="Email"/>
                                                    <xs:enumeration value="Calendar"/>
                                                    <xs:enumeration value="Contacts"/>
                                                    <xs:enumeration value="Tasks"/>
                                                    <xs:enumeration value="Notes"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                    </xs:all>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.11.1.1 Ping

The <Ping> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **Ping** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <HeartbeatInterval><br><Folders><br><MaxFolders> (response only)<br><Status> (response only) | **Container** | 1 (required) |

##### 2.2.2.11.1.1.1 HeartbeatInterval

The <HeartbeatInterval> element specifies the length of time, in seconds, that the server SHOULD wait before notifying the client of changes in a folder on the server. The <HeartbeatInterval> element is also returned by the server with a status code of 5 and specifies either the minimum or maximum interval that is allowed when the client has requested a heartbeat interval that is outside the acceptable range.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> | None | **Integer** | Request- 1 (Required in first request only)<br>Response- 0...1 (optional) |

The <HeartbeatInterval> element is only required in the first **Ping** command request from a device by a given user. The server then caches the heartbeat interval value so that in later requests the <HeartbeatInterval> element is necessary only if the client is changing the interval.

#### 2.2.2.11.1.1.2 Folders

The <Folders> element serves as a container for the <Folder> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> | <Folder> | **Container** | 0...1 (optional) |

##### 2.2.2.11.1.1.2.1 Folder

The <Folder> element contains the <Id> and <Class> elements in the **Ping** command request, which identifies the folder and folder type to be monitored by the client. The <Folder> element is also returned by the server with the <Status> element, where the element identifies the folder that is being described by the returned status code.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Folders> | <Id> (request only)<br><Class> (request only)<br>None (response only) | **Container** (request only)<br>**String** (response only) | 1...N (optional) |

###### 2.2.2.11.1.1.2.1.1 Id

The <Id> element specifies the server ID of the folder to be monitored.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Folder> (request only) | None | **String** (Up to 64 characters) | 1 (required) |

The server ID of the folder is obtained from the <folderhierarchy:ServerId> element of a previous **FolderSync** or **FolderCreate** command.

###### 2.2.2.11.1.1.2.1.2 Class

The <Class> element specifies the content class of the folder to be monitored. The possible content classes are *Email*, *Calendar*, *Contacts*, *Tasks*, and *Notes*.<19>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Folder> (request only) | None | **String** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.11.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **Ping** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="Ping:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="Ping:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="Ping">
          <xs:complexType>
               <xs:choice>
                    <xs:element name="Status" type="xs:unsignedByte" />
                    <xs:element minOccurs="0" name="Folders">
                         <xs:complexType>
                              <xs:sequence>
                                 <xs:element maxOccurs="unbounded" name="Folder" type="
xs:string "/>
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
                    <xs:element minOccurs="0" name="MaxFolders" type="xs:integer" />
                    <xs:element minOccurs="0" name="HeartbeatInterval" type="xs:integer" />
               </xs:choice>
          </xs:complexType>
     </xs:element>
</xs:schema>
```

##### 2.2.2.11.2.1 Ping

The <Ping> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **Ping** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Folders><br><HeartbeatInterval><br><MaxFolders> (response only)<br><Status> (response only) | **Container** | 1 (required) |

###### 2.2.2.11.2.1.1 Status

The <Status> element indicates the success or failure of the **Ping** command request. If the command failed, the <Status> element contains a code that indicates the type of failure. Certain status codes have additional information that is included in the response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> (response only) | None | **Unsigned byte** | 1 (required) |

The following table lists the <Status> codes for the **Ping** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | The heartbeat interval expired before any changes occurred in the folders being monitored. | | Global | Reissue the **Ping** command request. |
| 2 | Changes occurred in at least one of the monitored folders. The response specifies the changed folders. | | Global | Issue a **Sync** request for each folder that was specified in the **Ping** command response to retrieve the server changes. Reissue the **Ping** command when the **Sync** command completes to stay up to date. |
| 3 | The **Ping** command request omitted required parameters. | The **Ping** command request did not specify all the necessary parameters. The client MUST issue a **Ping** request that includes both the heartbeat interval and the folder list at least once. The server saves the heartbeat interval value (section 2.2.2.11.1.1.1), so only the folder list is required on subsequent requests. | Global | Reissue the **Ping** command request with the entire XML body. |
| 4 | Syntax error in **Ping** command request. | Frequently caused by poorly formatted WBXML. | Global | Fix bug in client code. |
| 5 | The specified heartbeat interval is outside the allowed range. For intervals that were too short, the response contains the shortest allowed interval. For intervals that were too long, the response contains the longest allowed interval. | The client sent a **Ping** command request with a heartbeat interval that was either too long or too short. | Global | Reissue the **Ping** command by using a heartbeat interval inside the allowed range. Setting the interval to the value returned in the **Ping** response will most closely accommodate the original value specified. |
| 6 | The **Ping** command request specified more than the allowed number of folders to monitor. The response indicates the allowed number in the <MaxFolders> element. | The client sent a **Ping** command request that specified more folders than the server is configured to monitor. | Global | Direct the user to select fewer folders to monitor. Resend the **Ping** command request with the new, shorter list. |
| 7 | Folder hierarchy sync required. | The folder hierarchy is out of date; a folder hierarchy sync is required. | Global | Issue a **FolderSync** command to get the new hierarchy and prompt the user, if it is |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| | | | | necessary, for new folders to monitor. Reissue the **Ping** command. |
| 8 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry. |

#### 2.2.2.11.2.1.2 Folders

The <Folders> element serves as a container for the <Folder> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> | <Folder> | **Container** | 0...1 (optional) |

##### 2.2.2.11.2.1.2.1 Folder

The <Folder> element contains the <Id> and <Class> elements in the **Ping** command request, which identifies the folder and folder type to be monitored by the client. The <Folder> element is also returned by the server with the <Status> element, where the element identifies the folder that is being described by the returned status code.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Folders> | <Id> (request only)<br><Class> (request only)<br>None (response only) | **Container** (request only)<br>**String** (response only) | 1...N (optional) |

#### 2.2.2.11.2.1.3 MaxFolders

The <MaxFolders> element specifies the maximum number of folders that can be monitored.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> (response only) | None | **Integer** | 0...1 (optional) |

The <MaxFolders> element is returned in a response with a status code of 6.

#### 2.2.2.11.2.1.4 HeartbeatInterval

The <HeartbeatInterval> element specifies the length of time, in seconds, that the server SHOULD wait before notifying the client of changes in a folder on the server. The <HeartbeatInterval> element is also returned by the server with a status code of 5 and specifies either the minimum or maximum interval that is allowed when the client has requested a heartbeat interval that is outside the acceptable range.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Ping> | None | **Integer** | Request- 1 (Required in first request only)<br>Response- 0...1 (optional) |

The <HeartbeatInterval> element is only required in the first **Ping** command request from a device by a given user. The server then caches the heartbeat interval value so that in later requests the <HeartbeatInterval> element is necessary only if the client is changing the interval.

### 2.2.2.12 Provision

The **Provision** command enables client devices to request from the server the security policy settings that the administrator sets, such as the minimum personal identification number (PIN) password length requirement. The **Provision** command is specified in [MS-ASPROV].

### 2.2.2.13 ResolveRecipients

The **ResolveRecipients** command is used by clients to resolve a list of supplied recipients, to retrieve their free/busy information, and, optionally, to retrieve their S/MIME certificates so that clients can send encrypted S/MIME e-mail messages.<20>

The ResolveRecipients namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ResolveRecipients namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.13.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **ResolveRecipients** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ResolveRecipients:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ResolveRecipients:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ResolveRecipients">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="To" maxOccurs="1000">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="256"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Options" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="CertificateRetrieval" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="3"/>
                  </xs:restriction>
```



Copyright © 2010 Microsoft Corporation.

```
            </xs:simpleType>
          </xs:element>
          <xs:element name="MaxCertificates" minOccurs="0">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="9999"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="MaxAmbiguousRecipients" minOccurs="0">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
                <xs:maxInclusive value="9999"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Availability" minOccurs="0">
            <xs:complexType>
              <xs:all>
                <xs:element name="StartTime" type="xs:string" />
                <xs:element name="EndTime" type="xs:string" minOccurs="0" />
              </xs:all>
            </xs:complexType>
          </xs:element>
          <xs:element name="Picture" minOccurs="0">
            <xs:complexType>
                <xs:all>
                    <xs:element name="MaxSize" type="xs:unsignedInt" minOccurs="0" />
                    <xs:element name="MaxPictures" type="xs:unsignedInt" minOccurs="0" />
                </xs:all>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:choice>
 </xs:complexType>
</xs:element>
</xs:schema>
```

#### 2.2.2.13.1.1 ResolveRecipients

The <ResolveRecipients> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **ResolveRecipients** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <To> (request only)<br><Options> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.13.1.1.1 To

The <To> element specifies one or more recipients to be resolved. The <To> element is also an **ambiguous name resolution (ANR)** search field.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ResolveRecipients> (request only)<br><Response> (response only) | None | **String**, limited to 256 characters. | 1...1000 (required) |

The result of including more than 1000 <To> elements in the request is undefined. The server MAY return a protocol status error in response to such a command request.

The server attempts to match the <To> value specified in the request to common directory service user attributes, and then return the matches. The <To> element(s) that are returned in the response corresponds directly to the <To> element(s) that are specified in the request.<21>

If the <To> element specifies an ambiguous name and the <Availability> element is included in the request, the response will not include free/busy data for that user. The Availability element is only included when <To> includes a valid SMTP address or name that resolves to a unique individual on the server.

If the **ResolveRecipients** command request includes the <Availability> element, a maximum of 100 <To> elements containing SMTP addresses can be included in the request. If more than 100 SMTP addresses are included in the request, <Status> value 160 is returned in the response.

If the **ResolveRecipients** command request includes the <Availability> element and the <To> element specifies a distribution group, then the availability data is returned as a single string that merges the data for the individual members of the distribution group. If the distribution group contains more than 20 members, a <Status> value of 161 is returned in the response indicating that the merged free busy information of such a large distribution group is not useful. For more information about the <Status> element, see section 2.2.2.13.2.1.1.

#### 2.2.2.13.1.1.2 Options

The <Options> element contains the options for resolving the list of recipients.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ResolveRecipients> (request only) | <CertificateRetrieval> (request only)<br><MaxCertificates> (request only)<br><MaxAmbiguousRecipients> (request only)<br><Availability> (request only)<br>Picture (request only)<22> | **Container** | 0...1 (optional) |

##### 2.2.2.13.1.1.2.1 CertificateRetrieval

The <CertificateRetrieval> element specifies whether S/MIME certificates SHOULD be returned by the server for each resolved recipient.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Integer** | 0...1 (optional) |

The following table shows valid values for the <CertificateRetrieval>.

| Value | Meaning |
|---|---|
| 1 | Do not retrieve certificates for the recipient (default). |
| 2 | Retrieve the full certificate for each resolved recipient. |
| 3 | Retrieve the mini certificate for each resolved recipient. |

#### 2.2.2.13.1.1.2.2 MaxCertificates

The <MaxCertificates> element limits the total number of certificates that is returned by the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | <Integer> | 0...1 |

The value of <MaxCertificates> is limited to a range of 0–9999. This limit ensures that no individual recipient receives an incomplete set of certificates. If the <MaxCertificates> limit is reached while enumerating certificates for an **address list**, that address list won't get back any certificates and a <Status> value of 8 is returned. The client can then use the certificate count returned to determine the number of certificates that are available for that recipient node.

#### 2.2.2.13.1.1.2.3 MaxAmbiguousRecipients

The <MaxAmbiguousRecipients> element limits the number of suggestions that are returned for each ambiguous recipient node in the response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Integer** | 0...1 |

The value of <MaxAmbiguousRecipients> is limited to a range of 0–9999. Each ambiguous recipient node receives only this many suggestions and no more. The recipient count, returned in the <RecipientCount> element, can be used by the client to determine the total number of suggestions available for that recipient.

#### 2.2.2.13.1.1.2.4 Availability

The <Availability> element indicates to the server that free/busy data is being requested by the client. The <Availability> element identifies the start time and end time of the free/busy data to retrieve.<23>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Recipient> (response only) | <StartTime> (request only)<br><EndTime> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <MergedFreeBusy> (response only) | | |

When the <Availability> element is included in a **ResolveRecipients** request, the server retrieves free/busy information for the users identified in the <To> elements included in the request, and returns the free/busy information in the <MergedFreeBusy> element in the response. If the <Availability> element is included in the **ResolveRecipients** request, the request must also include a valid <StartTime> and <EndTime>. When the server parses the request, the server first resolves the recipients identified by the <To> elements, and then determines the users free/busy information for the specified time span, before returning the free/busy data in the <MergedFreeBusy> element.

#### 2.2.2.13.1.1.2.4.1 StartTime

The <StartTime> element identifies the start of the span of free/busy time requested by the client.<24>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Availability> (request only) | None | **DateTime** | 1...1 (required) |

If the <Availability> element is included in the request, the request MUST also include the <StartTime> and <EndTime> elements.

If the client sends an invalid <StartTime> value, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

#### 2.2.2.13.1.1.2.4.2 EndTime

The <EndTime> element identifies the end of the span of free/busy time requested by the client.<25>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Availability> (request only) | None | **DateTime** | 1...1 (required) |

If the <Availability> element is included in the request, the request MUST also include the <StartTime> and <EndTime> elements.

If the client sends an invalid <EndTime> value, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

The result of including no <EndTime> element in an <Availablity> request is undefined. The server MAY return a protocol status error in response to such a command request.

If the <EndTime> value specified in the request is smaller than the <StartTime> value plus 30 minutes, or the duration spanned by the <StartTime> and the <EndTime> is greater than 42 days, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

#### 2.2.2.13.1.1.2.5 Picture

The inclusion of the <Picture> element<26> in a **ResolveRecipients** command request identifies that the client is requesting that contact photos be returned in the **ResolveRecipients** command response.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Properties> (response only) | <MaxSize> (request only)<br><MaxPictures> (request only)<br><Status> (response only)<br><Data> (response only) | **Container** (request or response)<br>**Empty** (request only) | 0…1 (optional) |

#### 2.2.2.13.1.1.2.5.1 MaxSize

The <MaxSize> element<27> limits the size of contact photos returned in the **ResolveRecipients** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> | None | **Integer** | 0…1 (optional) |

The maximum value of the <MaxSize> element is 100 KB or 102400 bytes.

The <MaxSize> element specifies the maximum size of an individual contact photo that is returned in the response, in bytes. The <MaxPictures> element specifies the maximum number of contact photos to return.

#### 2.2.2.13.1.1.2.5.2 MaxPictures

The <MaxPictures> element<28> limits the number of contact photos returned in a **ResolveRecipients** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> | None | **Integer** | 0…..1 (optional) |

The server returns the first N results that have contact photos, where N is the value of the <MaxPictures> element. After the <MaxPictures> limit is reached, the server returns <Status> value 173 (NoPicture) if the contact has no photo, or <Status> value 175 (PictureLimitReached) if the contact has a photo but the <MaxPictures> limit was reached.

Note that the <MaxPictures> element identifies the number of contact photos returned per query. Therefore, if the client includes three recipients to resolve and sets <MaxPictures> to 3, a maximum of 9 contact photos can be returned.

### 2.2.2.13.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **ResolveRecipients** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ResolveRecipients:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ResolveRecipients:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ResolveRecipients">
   <xs:complexType>
      <xs:choice>
         <xs:element name="Status" type="xs:unsignedByte" />
```



Copyright © 2010 Microsoft Corporation.

```
          <xs:element minOccurs="0" name="Response">
             <xs:complexType>
             <xs:sequence>
                <xs:element name="To">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:maxLength value="256"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="Status" type="xs:string"/>
                <xs:element name="RecipientCount" type="xs:integer"/>
                <xs:element maxOccurs="unbounded" name="Recipient">
                   <xs:complexType>
                   <xs:sequence>
                      <xs:element name="Type" type="xs:unsignedByte"/>
                      <xs:element name="DisplayName" type="xs:string"/>
                      <xs:element name="EmailAddress" type="xs:string"/>
                      <xs:element minOccurs="0" name="Availability"/>
                         <xs:complexType>
                         <xs:sequence>
                            <xs:element name="Status" type="xs:string"/>
                            <xs:element minOccurs="0" name="MergedFreeBusy" type="xs:string"/>
                         </xs:sequence>
                         </xs:complexType>
                      </xs:element>
                      <xs:element minOccurs="0" name="Certificates">
                         <xs:complexType>
                         <xs:sequence>
                            <xs:element name="Status" type="xs:unsignedByte"/>
                            <xs:element name="CertificateCount" type="xs:integer"/>
                            <xs:element name="RecipientCount" type="xs:integer"/>
                            <xs:element minOccurs="0" maxOccurs="unbounded" name="Certificate"
type="xs:string"/>
                            <xs:element minOccurs="0" name="MiniCertificate"
type="xs:string"/>
                         </xs:sequence>
                         </xs:complexType>
                      </xs:element>
                      <xs:element minOccurs="0" maxOccurs="unbounded" name="Pictures">
                        <xs:complexType>
                        <xs:all>
                          <xs:element name="Status" type="xs:string"/>
                          <xs:element name="Data" type="xs:string"/>
                        </xs:all>
                        </xs:complexType>
                      </xs:element>
                   </xs:sequence>
                   </xs:complexType>
                </xs:element>
             </xs:sequence>
             </xs:complexType>
          </xs:element>
       </xs:choice>
    </xs:complexType>
  </xs:element>
  </xs:schema>
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.13.2.1 ResolveRecipients

The <ResolveRecipients> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **ResolveRecipients** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <To> (request only)<br><Options> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1 (required) |

##### 2.2.2.13.2.1.1 Status

The <Status> element provides the status of the **ResolveRecipients** response. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ResolveRecipients> (response only) | None | **Unsigned byte** | 1 |

The following table shows valid values for the <Status> element when it is returned as a child of the <ResolveRecipients> element.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 5 | Protocol error. Either an invalid parameter was specified or the range exceeded limits. |
| 6 | An error occurred on the server. The client SHOULD retry the request. |

##### 2.2.2.13.2.1.2 Response

The <Response> element contains information as to whether the recipient was resolved; if the recipient was resolved, the element also contains the type of recipient, the e-mail address that the recipient resolved to, and, optionally, the S/MIME certificate for the recipient.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ResolveRecipients> (response only) | <To> (response only)<br><Status> (response only)<br><RecipientCount> (response only)<br><Recipient> (response only) | **Container** | 0…1 (optional) |

###### 2.2.2.13.2.1.2.1 To

The <To> element specifies a recipient to be resolved and is an ANR search field.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ResolveRecipients> (request only)<br><Response> (response only) | None | **String**, limited to 256 characters. | 1...1 (required) |

The server attempts to match the <To> value specified in a request to common directory service user attributes, and then return the matches. The <To> element(s) that are returned in the response correspond directly to the <To> element(s) that are specified in the request.<29>

If the request message includes the <Availability> element and includes a <To> element for an ambiguous user, the response does not include a <MergedFreeBusy> element for that user. Only users or **distribution lists** specified with valid SMTP addresses or a uniquely identifiable string in the request message <To> element have <MergedFreeBusy> elements included in the response.

#### 2.2.2.13.2.1.2.2 Status

The <Status> element provides the status of the **ResolveRecipient** <Response> element. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | None | **Unsigned byte** | 1 |

The following table shows valid values for the <Status> element when it is returned as a child of the <Response> element.

| Value | Meaning |
|---|---|
| 1 | The recipient was resolved successfully. For more details about the <Recipient> element, see section 2.2.2.13.2.1.2.4. |
| 2 | The recipient was found to be ambiguous. The returned list of recipients are suggestions. No certificate nodes were returned. Prompt the user to select the intended recipient from the list returned. |
| 3 | The recipient was found to be ambiguous. The returned list is a partial list of suggestions. The total count of recipients can be obtained from the <RecipientCount> element. No certificate nodes were returned. Prompt the user to select the intended recipient from the list returned or to get more recipients. |
| 4 | The recipient did not resolve to any contact or **Global Address List (GAL)** entry. No certificates were returned. Inform the user of the error and direct the user to check the spelling. |

#### 2.2.2.13.2.1.2.3 RecipientCount

The <RecipientCount> element specifies the number of recipients that are returned in the **ResolveRecipients** command response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | None | **Integer** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.13.2.1.2.4 Recipient

The <Recipient> element represents a single recipient that has been resolved.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Response> (response only) | <Type> (response only)<br><DisplayName> (response only)<br><EmailAddress> (response only)<br><Availability> (response only)<br><Certificates> (response only)<br><Picture> (response only) | **Container** | 0...N |

One or more <Recipient> elements are returned to the client in a <Response> element by the server if the <To> element specified in the request was either resolved to a distribution list or found to be ambiguous. The status code returned in the <Response> element can be used to determine if the recipient was found to be ambiguous. The recipient would be a suggested match if the recipient specified in the request was found to be ambiguous.

A <Certificates> element is returned as a child of <Recipient> if the client requested certificates to be returned in the response.

##### 2.2.2.13.2.1.2.4.1 Type

The <Type> element indicates the type of recipient, either a contact entry (2) or a GAL entry (1).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Recipient> (response only) | None | Unsigned byte | 1...1 (required) |

##### 2.2.2.13.2.1.2.4.2 DisplayName

The <DisplayName> element contains the display name of the recipient.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Recipient> (response only) | None | **String** | 1 per <Recipient> parent element |

##### 2.2.2.13.2.1.2.4.3 EmailAddress

The <EmailAddress> element contains the e-mail address, in SMTP format, of the recipient.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Recipient> (response only) | None | **String** | 1 per <Recipient> parent element |

##### 2.2.2.13.2.1.2.4.4 Availability

The <Availability> element returns the status and free/busy data of the users or distribution lists identified in the request for the time identified by the <StartTime> and <EndTime> elements.<30>



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Recipient> (response only) | <StartTime> (request only)<br><EndTime> (request only)<br><Status> (response only)<br><MergedFreeBusy> (response only) | **Container** | 0...1 (optional) |

#### 2.2.2.13.2.1.2.4.4.1 Status

The <Status> element provides the status of the **ResolveRecipient** <Availability> element. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <>Availability (response only) | None | **Unsigned byte** | 1 |

The following table shows valid values for the <Status> element when it is returned as a child of the <Availability> element.<31>

| Value | Meaning |
|---|---|
| 1 | Free/busy data was successfully retrieved for a given recipient. This value does not indicate that the response is complete. |
| 160 | There were more than 100 recipients identified by the <To> elements in the **ResolveRecipient** request. |
| 161 | The distribution group identified by the <To> element of the **ResolveRecipient** request included more than 20 recipients. |
| 162 | The free/busy data could not be retrieved by the server due to a temporary failure. The client SHOULD reissue the request. This error is caused by a timeout value being reached while requesting free/busy data for some users, but not others. |
| 163 | Free/busy data could not be retrieved from the server for a given recipient. Clients SHOULD NOT reissue the request as it is caused by a lack of permission to retrieve the data. |

#### 2.2.2.13.2.1.2.4.4.2 MergedFreeBusy

The <MergedFreeBusy> element contains a string that identifies the free/busy information for the users or distribution list identified in the request.<32>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Availability> (response only) | None | **String** | 0...1 (optional) |

The <MergedFreeBusy> string has a maximum length of 32KB. To retrieve more than 32KB of availability data, the client MUST issue a new request with the appropriate start time and end time.

Each digit in the <MergedFreeBusy> string indicates the free/busy status for the user or distribution list for every 30 minute interval. The following table lists the valid values:



Copyright © 2010 Microsoft Corporation.

| Digit | Availability |
|---|---|
| 0 | Free |
| 1 | Tentative |
| 2 | Busy |
| 3 | **Out of Office (OOF)** |
| 4 | No data |

A string value of "32201" would represent that this user or group of users is out of the office for the first 30 minutes, busy for the next hour, free for 30 minutes, and then has a tentative meeting for the last 30 minutes. If the user or group of users have a change in availability that lasts less than the interval value of 30 minutes, the availability value with the higher digit value is assigned to the whole interval period. For example, if a user has a 25 minutes of free time (value 0) followed by 5 minutes of busy time (value 2), the 30 minute interval is assigned a value of 2 in the server response.

The server determines the number of digits to include in the <MergedFreeBusy> element by dividing the time interval specified by <StartTime> and <EndTime> by 30 minutes, and rounding the result up to the next integer.

The <MergedFreeBusy> string is populated from the <StartTime> onwards, therefore the last digit represents between a millisecond and 30 minutes. A query for data from 13:00:00 to 13:30:00 returns a single digit but a query from 12:59:59 to 13:30:00 or 13:00:00 to 13:30:01 returns two digits.

Any appointment that ends inside a second of the interval requested shall impact the digit representing that timeframe. For example, given a calendar that contains a 5 minute OOF appointment from 12:00 to 12:05, and is free the rest of the day, queries would result in the following:

- If a query is made for 12:00:00 to 13:00:00, the result is "30", where each digit represents exactly 30 minutes.
- If a query is made for 12:04:59 to 13:00:00, the result is "30", where the "0" maps to 12:34:59 to 13:00:00.
- If a query is made for 12:05:00 to 13:00:00, the result is "00" where the second 0 maps the last 25 minutes of the interval.

The client MUST consider daylight saving time transitions and may need to add or remove time intervals from the <MergedFreeBusy> string, as there are days that have more or less than 24 hours.

If the <Availability> element is included in the response, the response MUST also include the <Status> element. The <MergedFreeBusy> element is also included if the <Status> value indicates success.

#### 2.2.2.13.2.1.2.4.5 Certificates

The <Certificates> element contains information about the certificates for a recipient.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Recipient> (response only) | <Status> (response only)<br><CertificateCount> (response only)<br><RecipientCount> (response only)<br><Certificate> (response only)<br><MiniCertificate> (response only) | **Container** | 0...1 |

Although the multiple <Certificates> element can be included in a response, only one <Certificates> element is allowed per <Recipient> parent element.

#### 2.2.2.13.2.1.2.4.5.1 Status

The <Status> element provides the status of the **ResolveRecipient** <Certificates> element. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (response only) | None | **Unsigned byte** | 1 |

The following table shows valid values for the <Status> element when it is returned as a child of the <Certificates> element.

| Value | Meaning |
|---|---|
| 1 | One or more certificates were successfully returned. |
| 7 | The recipient does not have a valid S/MIME certificate. No certificates were returned. |
| 8 | The global certificate limit was reached and the recipient's certificate could not be returned. The count certificates not returned can be obtained from the <CertificateCount> element. Retry with fewer recipients if possible, otherwise prompt the user. |

#### 2.2.2.13.2.1.2.4.5.2 CertificateCount

The <CertificateCount> element specifies the number of valid certificates that were found for the recipient.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (response only) | None | **Integer** | 1 per <Certificates> parent element |

If a status code of 8 is returned with the <Certificates> element, the <CertificateCount> element specifies the number of recipient certificates that was not returned.

#### 2.2.2.13.2.1.2.4.5.3 RecipientCount

The <RecipientCount> element specifies the number of members belonging to a distribution list.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (response only) | None | **Integer** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

When returned in the <Certificates> element, the <RecipientCount> can be used to determine whether all recipients belonging to a distribution list have valid certificates by comparing values of the <CertificateCount> and <RecipientCount> elements.

#### 2.2.2.13.2.1.2.4.5.4 Certificate

The <Certificate> element contains the base64-encoded X509 certificate BLOB.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (response only) | None | **String** (base64 encoded) | 0...N |

This element is returned by the server only if the client specified a value of 2 in the <CertificateRetrieval> element in the request.

#### 2.2.2.13.2.1.2.4.5.5 MiniCertificate

The <MiniCertificate> element contains the base64-encoded mini-certificate BLOB.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (response only) | None | **String** (base64 encoded) | 0...1 per <Certificates> parent element |

This element is returned only if the client specified a value of 3 in the <CertificateRetrieval> element in the request.

#### 2.2.2.13.2.1.2.4.6 Picture

The <Picture> element<33> contains the data related to the contact photos.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Recipient> (response only) | <MaxSize> (request only)<br><MaxPictures> (request only)<br><Status> (response only)<br><Data> (response only) | **Container** (request or response<br>**Empty** (request only) | 0…1 (optional) |

#### 2.2.2.13.2.1.2.4.6.1 Status

The <Status> element provides the status of the **ResolveRecipient** <Picture> element. For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> (response only) | None | **Unsigned byte** | 1 |

The following table shows valid values for the <Status> element when it is returned as a child of the <Picture> element.<34>



Copyright © 2010 Microsoft Corporation.

| Value | Meaning |
|---|---|
| 1 | The contact photo was retrieved successfully. |
| 173 | The user does not have a contact photo. |
| 174 | The contact photo exceeded the size limit set by the <MaxSize> element (section 2.2.2.13.1.1.2.5.1). |
| 175 | The number of contact photos returned exceeded the size limit set by the <MaxPictures> element (section 2.2.2.13.1.1.2.5.2). |

#### 2.2.2.13.2.1.2.4.6.2 Data

The <Data> element<35> contains the binary data of the contact photo.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> (response only) | None | **Byte array** | 0…1 (optional) |

### 2.2.2.14 Search

The **Search** command is used to find entries in an **address book**, mailbox, or document library (UNC or Windows SharePoint Services).

The Search namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the Search namespace use the namespace prefixes defined in section 2.2.1.

The Accept-Language header in a **Search** command request is used to define the **locale** of the client so that the search is relevant. If the accept language is not specified, the search is conducted by using the server language.

Searching the Global Address List (GAL)

The **Search** command is used to find contacts and recipients in the GAL, and to retrieve information about them. When a search query matches more than one GAL entry, the **Search** command MUST return as many entries as requested, up to a total of 100 entries by default.

For each GAL entry that is found, the **Search** command returns all the non-empty properties that are indexed by the online ANR in the global catalog server—for example, e-mail **alias**, display name, first and last names, company name, and so on.

The client can optionally specify the maximum number of entries to retrieve in the **Search** command request by specifying the range. The server MUST return entries up to the number that is requested, and MUST also indicate the total number of entries that are found.

The text query string that is provided to the **Search** command is used in a prefix-string match. For example, if the client performs a **Search** with a <Query> element value of "Michael A.", the command returns the entries that contain the search string in any text field, such as "Michael Alexander", "Michael Allen". Because the **Search** command matches the <Query> element value against all ANR-indexed GAL text properties, the client can also search by e-mail address, company name, and so on.

The ANR system indexes the following properties:



Copyright © 2010 Microsoft Corporation.

- Display name
- Alias
- FirstName
- LastName
- EmailAddress

The **Search** command results are sorted by the server according to their ordering in the GAL (that is, by the display name property). Because of how the search results are sorted, the client could have to sort the results to display results in a relevant manner to users. For example, a search for "123" might return all GAL entries that have mailing addresses or e-mail addresses that begin with 123. The client can choose to display matching e-mail addresses before mailing addresses, if they know their users use e-mail addresses more frequently than mailing addresses, or mailing addresses before e-mail addresses if mailing addresses are used more frequently.

The <Range> option is a zero-based index specifier in the form of "m-n". For more details about the meaning of the <Range> values, see section 2.2.2.14.1.1.1.3.2.

Searching Outside the GAL

Typical Usage

Essentially, search involves the following three phases:

1. The client issues a request for specific search results.
2. The client uses subsequent requests to retrieve more results by incrementing the range.
3. Any actions on the search results are carried out by using other protocol commands, such as **ItemOperations**, **SmartReply**, or **SmartForward**.

The following figure shows the typical usage of the **Search** command to retrieve successive result sets from the server and then perform some action based on those results (for example, retrieve the full message body for an e-mail search result).



Copyright © 2010 Microsoft Corporation.

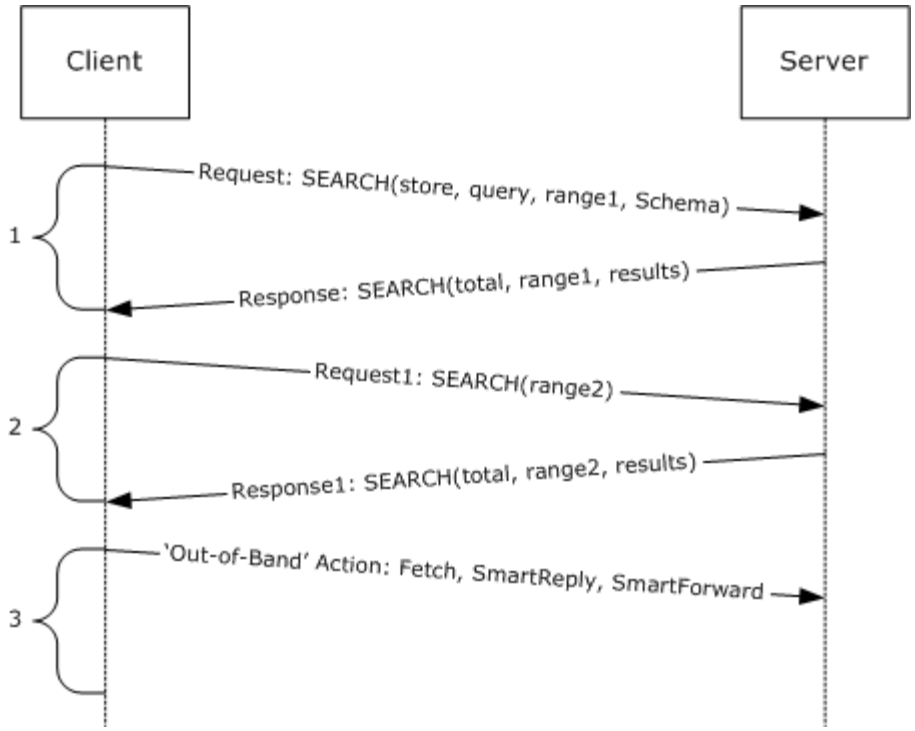


**Figure 2: Search command process**

#### 2.2.2.14.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **Search** command request.

The following **Search** command elements are specified in other documents:

- <airsyncbase:BodyPreference> ([MS-ASAIRS] section 2.2.2.2)
- <airsyncbase:BodyPartPreference> ([MS-ASAIRS] section 2.2.2.3)
- <email:DateRecieved> ([MS-ASEMAIL] section 2.2.2.6)
- <rm:RightsManagementSupport> ([MS-ASRM] section 2.2.2.1)

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        id="Search"
        targetNamespace="Search:"
        xmlns:email="Email:"
        xmlns:mstns="Search:"
        xmlns="Search:"
        xmlns:airsync="AirSync:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:documentlibrary="DocumentLibrary:"
        xmlns:rm="RightsManagement:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        attributeFormDefault="qualified"
        elementFormDefault="qualified">
```



Copyright © 2010 Microsoft Corporation.

```
<xs:import namespace="DocumentLibrary:"/>
<xs:import namespace="AirSync:"/>
<xs:import namespace="AirSyncBase:"/>
<xs:import namespace="Email:"/>
<xs:import namespace="RightsManagement:"/>

<xs:element name="LongId">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="256"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:complexType name="EmptyTag" />
<xs:complexType name="queryType" mixed="true">
    <xs:sequence>
        <xs:choice maxOccurs="unbounded">
            <xs:element name="And" type="queryType" minOccurs="0"/>
            <xs:element name="FreeText" type="xs:string" minOccurs="0"/>
            <xs:element ref="airsync:Class" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="airsync:CollectionId" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="ConversationId" type="xs:string" minOccurs="0"/>
            <xs:element name="EqualTo" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="documentlibrary:LinkId"/>
                        <xs:element name="Value">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="1024"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="GreaterThan" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="email:DateReceived"/>
                        <xs:element name="Value">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="1024"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="LessThan" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="email:DateReceived"/>
                        <xs:element name="Value">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="1024"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Search">
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="Store">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="Name">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="1"/>
                                        <xs:maxLength value="256"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="Query" type="queryType" />
                            <xs:element name="Options" minOccurs="0">
                                <!-- Must differentiate between document library and Mailbox
options...!-->
                                <xs:complexType>
                                    <xs:choice maxOccurs="unbounded">
                                        <xs:element ref="airsync:MIMESupport" minOccurs="0"/>
                                        <xs:element ref="airsyncbase:BodyPreference"
minOccurs="0" maxOccurs="unbounded" />
                                        <xs:element ref="airsyncbase:BodyPartPreference"
minOccurs="0"/>
                                        <xs:element ref="rm:RightsManagementSupport"
minOccurs="0"/>
                                        <xs:element name="Range">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:pattern value="[0-9]{1,3}-[0-
9]{1,3}"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="UserName">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="100" />
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="Password">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="100" />
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="DeepTraversal" type="EmptyTag" />
```



Copyright © 2010 Microsoft Corporation.

```
                                        <xs:element name="RebuildResults" type="EmptyTag" />
                                        <xs:element name="Picture" minOccurs="0">
                                            <xs:complexType>
                                                <xs:all>
                                                    <xs:element name="MaxSize"
type="xs:unsignedInt" minOccurs="0"/>
                                                    <xs:element name="MaxPictures"
type="xs:unsignedInt" minOccurs="0"/>
                                                </xs:all>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.14.1.1 Search

The <Search> element is the top-level element in the XML stream for the **Search** command. The element identifies the body of the HTTP **POST** as containing a **Search** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Store> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1...1 (required) |

##### 2.2.2.14.1.1.1 Store

In the **Search** command request XML, the <Store> element is a container for elements that specify the location, string, and options for the search. In the **Search** command response XML, the <Store> element contains the <Status>, <Result>, <Range>, and <Total> elements that contain the returned mailbox entries.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Search> (request only)<br><Response> (response only) | <Name> (request only)<br><Query> (request only)<br><Options> (request only)<br><Status> (response only)<br><Result> (response only)<br><Range> (response only)<br><Total> (response only) | **Container** | 1...1 (required) |

The result of including more than one <Store> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.14.1.1.1.1 Name

The <Name> element in the **Search** command request specifies which store to search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (request only) | None | **String** (up to 256 characters) | 1...1 (required) |

The <Search> element accepts the following values:

- Mailbox—The client specifies "Mailbox" when it intends to search the **message database**.
- Document Library—The client specifies "DocumentLibrary" when it intends to search a Windows SharePoint Services or UNC library.
- GAL – The client specifies "GAL" when it intends to search the Global Address List.

#### 2.2.2.14.1.1.1.2 Query

The <Query> element specifies the keywords to use for matching the entries in the store that is being searched.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (request only) | <And> (request only)<br><EqualTo> (request only) | **Container** | 1...1 (Required, request only) |

The value of the <Query> element is used as a prefix-string matching pattern, and returns entries that match the beginning of the string. For example, searching for "John" would match "John Frum" or "Barry Johnson", but would not match "James Littlejohn".

All nonempty ANR-indexed text properties in the GAL are compared with the <Query> element value. Search comparisons are performed by using case-insensitive matching.

For a Windows SharePoint Services document library search, this protocol supports queries of the following form: LinkId == value, where value specifies the URL of the item or folder and LinkId indicates that the value is to be compared to the link ID property.

For mailbox search, the query syntax is as follows:

- Folders can be specified in the following ways:
  - Specified ID
  - Specified folder and subfolders
  - All e-mail folders, including Draft, Inbox and subfolders, Outbox, and Sent Items
- The basic keyword query can be composed of the following:
  - The basic operator: <And>
  - A **dateTime** filter specified by using the <GreaterThan> and <LessThan> elements
  - <FreeText> elements that contain keywords

The basic keyword query is executed against all indexed properties.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.14.1.1.1.2.1 And

The <And> element is a container that specifies items on which to perform an AND operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Query> (request only) | <FreeText> (request only)<br><airsync:Class> (request only)<br><airsync:CollectionId> (request only)<br><ConversationId> (request only)<br><GreaterThan> (request only)<br><LessThan> (request only) | **Container** | 0...1 (Optional, request only) |

If the <And> element is included as a child of any element other than <Query>, or if multiple <And> elements are included in the request, the server responds with a <Status> value of 8 (SearchTooComplex).

##### 2.2.2.14.1.1.1.2.1.1 FreeText

The <FreeText> element specifies a string for which to search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only) | None | **String** | 0...1 (Optional, request only) |

If the <FreeText> element is included as a child of any element other than the <And> element, the server responds with a <Status> value of 8 (SearchTooComplex).

##### 2.2.2.14.1.1.1.2.1.2 airsync:Class

The <airsync:Class> element specifies the classes that the client wants returned for a given collection. It is defined as an element in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only)<br><Schema> (request only)<br><Result> (response only) | None | **String** | 0...N (optional) |

The valid <airsync:Class> element values are:

- Tasks
- Email
- Calendar
- Contacts
- Notes
- SMS<36>



Copyright © 2010 Microsoft Corporation.

The **Search** request can include one or more <airsync:Class> elements in the request to limit the type of data included in the **Search** response.

If one or more <airsync:Class> elements are not included in the **Search** request, the server will return all supported classes.

If the <airsync:Class> element is included as a child of any element other than the <And> element, the server responds with a <Status> value of 8 (SearchTooComplex).

#### 2.2.2.14.1.1.1.2.1.3 airsync:CollectionId

The <airsync:CollectionId> element specifies the folder in which to search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only)<br><Result> (response only) | None | **String** | 0...N (optional) |

Multiple folders can be specified by including multiple <airsync:CollectionId> elements.

If the <DeepTraversal> element is present, it applies to all folders under each <airsync:CollectionId>.

If the <airsync:CollectionId> element is included as a child of any element other than <And>, the server responds with a <Status> value of 8 (SearchTooComplex).

#### 2.2.2.14.1.1.1.2.1.4 ConversationId

The <ConversationId> element specifies the conversation for which to search.<37> The value is a **GUID**. For details, see [MS-ASCON].

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only) | None | **String** | 0...1 (optional) |

If the <ConversationId> element is included as a child of any element other than <And>, the server responds with a <Status> value of 8 (SearchTooComplex).

The result of including more than one <ConversationId> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.14.1.1.1.2.1.5 GreaterThan

The <GreaterThan> element is a container that specifies a property and a value that are compared for a "greater than" **condition** during a search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only) | <email:DateReceived> (request only)<br><Value> (request only) | **Container** | 0...1 (optional, Request only) |

The result of including more than one <GreaterThan> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.



Copyright © 2010 Microsoft Corporation.

The <GreaterThan> element is supported only in mailbox searches. It is not supported for document library searches. The comparison is made between the value of the <Value> element and the date that a mailbox item was received. The <email:DateRecieved> element MUST be present before the <Value> element.

Typically, this element is used to filter results by the date on which they were received so that the date received is greater than the specified value.

If the <GreaterThan> element is included as a child of any element other than <And>, the server responds with a <Status> value of 8 (SearchTooComplex).

#### 2.2.2.14.1.1.1.2.1.5.1 email:DateReceived

The <email:DateReceived> element specifies the date that a mailbox item was received.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <GreaterThan> (request only) | None | **DateTime** | 1 (required) |

#### 2.2.2.14.1.1.1.2.1.5.2 Value

The <Value> element specifies the value that is to be used in a comparison.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <GreaterThan> (request only) | None | **String** (1,024 bytes maximum length) | 0...1 (Optional, request only) |

The <Value> element is used in the query together with an element that specifies the name of a property. The value that is specified by the <Value> element is compared with the value of the specified property.

#### 2.2.2.14.1.1.1.2.1.6 LessThan

The <LessThan> element is a container that specifies a property and a value that are compared for a "less than" condition during a search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only) | <email:DateReceived> (request only)<br><Value> (request only) | **Container** | 0...1 (Optional, request only) |

The result of including more than one <LessThan> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The <LessThan> element is supported only in mailbox searches. It is not supported for document library searches. The comparison is made between the value of the <Value> element and the date that a mailbox item was received. The <email:DateReceived> element MUST be present before the <Value> element.

Typically, this element is used to filter results by the date on which they were received so that the date received is less than the specified value.



Copyright © 2010 Microsoft Corporation.

If the <LessThan> element is included as a child of any element other than <And>, the server responds with a <Status> value of 8 (SearchTooComplex).

#### 2.2.2.14.1.1.1.2.1.6.1 email:DateReceived

The <email:DateReceived> element specifies the date that a mailbox item was received.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <LessThan> (request only) | None | **DateTime** | 1 (required) |

#### 2.2.2.14.1.1.1.2.1.6.2 Value

The <Value> element specifies the value that is to be used in a comparison.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <LessThan> (request only) | None | **String** (1,024 bytes maximum length) | 0...1 (Optional, request only) |

The <Value> element is used in the query together with an element that specifies the name of a property. The value that is specified by the <Value> element is compared with the value of the specified property.

### 2.2.2.14.1.1.1.2.2 EqualTo

The <EqualTo> element is a container that specifies a property and a value that are compared for equality during a search.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Query> (request only) | <documentlibrary:LinkId> (request only)<br><Value> (request only) | **Container** | 0...1 (optional, Request only) |

The result of including more than one <EqualTo> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The <Query> element is only supported as a parent element in a document library search.

The comparison is made between the value of the <Value> element and the <documentlibrary:LinkId> element.

#### 2.2.2.14.1.1.1.2.2.1 documentlibrary:LinkId

The <documentlibrary:LinkId> element specifies a URI that identifies a resource.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EqualTo> | None | **URI** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

The link ID is a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or UNC documents. The client MUST store the link ID that is associated with the items that are retrieved by using the **Search** command if it wants to act upon them later.

#### 2.2.2.14.1.1.1.2.2.2 Value

The <Value> element specifies the value that is to be used in a comparison.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EqualTo> (request only) | None | **String** (1,024 bytes maximum length) | 0...1 (Optional, request only) |

The <Value> element is used in the query together with an element that specifies the name of a property. The value that is specified by the <Value> element is compared with the value of the specified property.

#### 2.2.2.14.1.1.1.3 Options

The <Options> element is a container for search options.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (request only) | <airsync:MIMESupport> (request only)<br><airsyncbase:BodyPreference> (request only) as specified in [MS-ASAIRS] section 2.2.2.2<br><airsyncbase:BodyPartPreference> (request only) as specified in [MS-ASAIRS] section 2.2.2.3<br><rm:RightsManagementSupport> (request only) as specified in [MS-ASRM] section 2.2.2.1<br><Range> (request only)<br><UserName> (request only)<br><Password> (request only)<br><DeepTraversal> (request only)<br><RebuildResults> (request only)<br><Picture> (request only) | **Container** | 0...1 (optional, request only) |

The <UserName> and <Password> can only be sent in the request after receiving a status 14 (see section 2.2.2.14.2.1.1 for more details). The server requires these credentials to access the requested resources. The client MUST only send these over a secure or trusted connection, and only in response to a status 14. <UserName> and <Password> are defined as strings consisting of at most 100 characters.

The supported options vary according to the store that is being searched. The following table lists the valid options for each store.

| Options | Store |
|---|---|
| <Range><br><UserName> | GAL |



Copyright © 2010 Microsoft Corporation.

| Options | Store |
|---|---|
| <Password><br><Picture> | |
| <Range><br><DeepTraversal><br><RebuildResults><br><airsyncbase:BodyPreference><br><airsyncbase:BodyPartPreference><br><rm:RightsManagementSupport> | Mailbox |
| <Range><br><UserName><br><Password> | Document Library |

#### 2.2.2.14.1.1.1.3.1 airsync:MIMESupport

The <airsync:MIMESupport> element is included in the <Options> element of a client **Search** command request to enable MIME support for e-mail items that are sent from the server to the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Integer** | 0...1 (optional) |

The result of including more than one <airsync:MIMESupport> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The following table lists the valid values for the element.

| Value | Meaning |
|---|---|
| 0 | Never send MIME data. |
| 1 | Send MIME data for S/MIME [RFC5751] messages only. Send regular body (non S/MIME) data for all other messages. |
| 2 | Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution. |

The **Search** response can include the S/MIME BLOB of a signed/encrypted message.

If the <airsync:MIMESupport> element is set to 1 or 2 in the **Search** request:

- The child of the <airsyncbase:BodyPreference> element, the <Type> element, SHOULD be included in the **Search** request, containing a value of 4 to inform the server that the device can read the MIME BLOB.
- The response from the server MUST include the <airsyncbase:Body> element, which is a child of the <Properties> element. The <airsyncbase:Body> element is a complex element and MUST contain the following child nodes in an S/MIME search response:



Copyright © 2010 Microsoft Corporation.

- The <airsyncbase:Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <airsyncbase:EstimatedDataSize> element to specify the rough total size of the data.
- The <airsyncbase:Truncated> element to indicate whether the MIME BLOB is truncated.
- The <airsyncbase:Data> element that contains the full MIME BLOB.

For more details about the <airsyncbase:Body> element or the <airsyncbase:BodyPreference> element, see [MS-ASAIRS] section 2.2.2.4 or 2.2.2.2, respectively.

#### 2.2.2.14.1.1.1.3.2 Range

The <Range> element is used in both the request and response XML documents. In the request XML, the <Range> element specifies the maximum number of matching entries to return. In the response XML, the <Range> element specifies the number of matching entries that are being returned.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Store> (response only) | None | Zero-based range in the form m-n | 0...1 (optional) |

The result of including more than one <Range> element is undefined. The server MAY return a protocol status error in response to such a command request.

The <Range> element value specifies a number of entries, but indicates different things depending on whether the element is in the request or the response XML.

The format of the <Range> element value is in the form of a zero-based index specifier, formed with a zero, a hyphen, and another numeric value: "m-n." The m indicates the lowest index of a zero-based array that would hold the items. The n indicates the highest index of a zero-based array that would hold the items. For example, a <Range> element value of 0–9 indicates 10 items, and 0–10 indicates 11 items. A <Range> element value of 0–0 indicates 1 item.

If the request does not include a <Range> element, the default <Range> value for each <Store> type is used. The following table identifies the default <Range> values and maximum results returned for each <Store> type:

| Store value | Default Range value | Maximum results returned |
|---|---|---|
| Mailbox | 0-99 | 100 |
| DocumentLibrary | 0-999 | 1000 |
| GAL | 0-99 | 100 |

If the <Range> value specified in the request exceeds the default range value, a <Status> value of 12 is returned to indicate that the maximum range has been exceeded, as specified in section 2.2.2.14.2.1.2.1.1.

In the request XML, the <Range> element value specifies the maximum number of entries to be returned to the client.



Copyright © 2010 Microsoft Corporation.

In the response XML, the <Range> element value specifies the actual number of entries that are returned in the response. The <Total> element in the response XML indicates an estimate of the total number of entries that matched the <Query> element value.

Search results are stored in a **search folder** on the server. This way, when a client comes back with the same query but a new row range, rows are pulled from the result set that is currently stored in the search folder. The entire result set does not have to be rebuilt.

#### 2.2.2.14.1.1.1.3.3 Username

The <Username> element is included in the <Options> element of a client **Search** command request to enable searches on usernames.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **String** (100 character maximum length) | 0…1 (optional) |

#### 2.2.2.14.1.1.1.3.4 Password

The <Username> element is included in the <Options> element of a client **Search** command request to enable searches on usernames.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **String** (100 character maximum length) | 0…1 (optional) |

#### 2.2.2.14.1.1.1.3.5 DeepTraversal

The <DeepTraversal> element indicates that the client wants the server to search all subfolders for the folders that are specified in the query.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Empty tag** | 0...1 (optional) |

The result of including more than one <DeepTraversal> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The <DeepTraversal> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <DeepTraversal/> tag.

If the <DeepTraversal> element is not present, the subfolders are not searched.

#### 2.2.2.14.1.1.1.3.6 RebuildResults

The <RebuildResults> element is used within the **Search** request to force the server to rebuild the search folder that corresponds to a given query.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Empty tag** | 0...1 (optional) |

The <RebuildResults> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <RebuildResults/> tag.

The result of including more than one <RebuildResults> element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The search results (that is, the result set) are stored in a search folder on the server. This way, when a client comes back with the same query but a new row range, rows are pulled from the result set that is currently stored in the search folder. The entire result set does not have to be rebuilt.

The search folder remains unchanged until the client does one of the following to update the result set:

- Sends a **Search** request, specifying a new query. In this case, the search folder is automatically rebuilt. The <RebuildResults> node does not have to be included.
- Sends a **Search** request that includes the <RebuildResults> node. In this case, the server is forced to rebuild the search folder.

If a new item is added, the item does not appear in the result set until the result set is updated. If an item is deleted, the server will filter the deleted item out of the result set.

The client SHOULD send a new **Search** request with the given query and include the <RebuildResults> option every few days to ensure accurate results for that query.

#### 2.2.2.14.1.1.1.3.7 Picture

The <Picture> element<38> identifies that the client is requesting that contact photos be returned in the **Search** command response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | <MaxSize> (request only)<br><MaxPictures> (request only) | **Container** or **Empty tag** | 0…1 (optional) |

#### 2.2.2.14.1.1.1.3.7.1 MaxSize

The <MaxSize> element<39> limits the size of contact photos returned in the **Search** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> (request only) | None | **Integer** | 0…1 (optional) |

The maximum value of the <MaxSize> element is 100 KB or 102400 bytes.

The <MaxSize> element specifies the maximum size of an individual contact photo that is returned in the response, in bytes. The <MaxPictures> element specifies the maximum number of contact photos to return.



Copyright © 2010 Microsoft Corporation.

### 2.2.2.14.1.1.1.3.7.2 MaxPictures

The <MaxPictures> element<40> limits the number of contact photos returned in the **Search** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Picture> (request only) | None | **Integer** | 0…..1 (optional) |

The server will return the first N results that have contact photos, where N is the value of the <MaxPictures> element. After the <MaxPictures> limit is reached, the server returns <Status> value 173 (NoPicture) if the contact has no photo, or <Status> value 175 (PictureLimitReached) if the contact has a photo but the <MaxPictures> limit was reached.

## 2.2.2.14.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **Search** command response schema.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="Search:"
   xmlns:airsyncbase="AirSyncBase:"
   xmlns:airsync="AirSync:"
   xmlns:email="Email:"
   xmlns:email2="Email2:"
   xmlns:calendar="Calendar:"
   xmlns:contacts="Contacts:"
   xmlns:contacts2="Contacts2:"
   xmlns:documentlibrary="DocumentLibrary:"
   xmlns:notes="Notes:"
   xmlns:tasks="Tasks:"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="Search:"
   elementFormDefault="unqualified"
   attributeFormDefault="unqualified">
   <xs:import namespace="AirSync:"/>
   <xs:import namespace="AirSyncBase:"/>
   <xs:import namespace="Email:"/>
   <xs:import namespace="Email2:"/>
   <xs:import namespace="Contacts:"/>
   <xs:import namespace="Contacts2:"/>
   <xs:import namespace="Calendar:"/>
   <xs:import namespace="DocumentLibrary:"/>
   <xs:import namespace="Notes:"/>
   <xs:import namespace="Tasks:"/>
   <xs:element name="Search">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Status" minOccurs="0"/>
            <xs:element name="Response">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="Store">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Status"/>
                              <xs:element name="Result" maxOccurs="unbounded">
                                 <xs:complexType>
```



Copyright © 2010 Microsoft Corporation.

```
                                    <xs:sequence>
                                       <xs:element ref="airsync:Class"/>
                                       <xs:element name="LongId" minOccurs="0">
                                          <xs:simpleType>
                                             <xs:restriction base="xs:string">
                                                <xs:minLength value="0"/>
                                                <xs:maxLength value="265"/>
                                             </xs:restriction>
                                          </xs:simpleType>
                                       </xs:element>
                                       <xs:element ref="airsync:CollectionId"/>
                                       <xs:element name="Properties">
                                          <xs:complexType>
                                             <xs:all>
                                                <xs:element ref="email:To" minOccurs="0"/>
                                                <xs:element ref="email:Cc" minOccurs="0"/>
                                                <xs:element ref="email:From" minOccurs="0"/>
                                                <xs:element ref="email:Subject"
  minOccurs="0"/>
                                                <xs:element ref="email:ReplyTo"
  minOccurs="0"/>
                                                <xs:element ref="email:DateReceived"
  minOccurs="0"/>
                                                <xs:element ref="email:DisplayTo"
  minOccurs="0"/>
                                                <xs:element ref="email:ThreadTopic"
  minOccurs="0"/>
                                                <xs:element ref="email:Importance"
  minOccurs="0"/>
                                                <xs:element ref="email:Read" minOccurs="0"/>
                                                <xs:element ref="airsyncbase:Attachments"
  minOccurs="0"/>
                                                <xs:element ref="airsyncbase:Body"
  minOccurs="0"/>
                                                <xs:element ref="airsyncbase:BodyPart"
  minOccurs="0"/>
                                                <xs:element ref="email:MessageClass"
  minOccurs="0"/>
                                                <xs:element ref="email:MeetingRequest"
  minOccurs="0"/>
                                                <xs:element ref="email:InternetCPID"
  minOccurs="0"/>
                                                <xs:element ref="email:Flag" minOccurs="0"/>
                                                <xs:element ref="email:NativeBodyType"
  minOccurs="0"/>
                                                <xs:element ref="email:ContentClass"
  minOccurs="0"/>
                                                <xs:element ref="email2:UmCallerID"
  minOccurs="0"/>
                                                <xs:element ref="email2:UmUserNotes"
  minOccurs="0"/>
                                                <xs:element ref="email2:ConversationId"
  minOccurs="0"/>
                                                <xs:element ref="email2:ConversationIndex"
  minOccurs="0"/>
                                                <xs:element ref="email2:Sender"
  minOccurs="0"/>
                                                <xs:element ref="email2:LastVerbExecuted"
  minOccurs="0"/>
                                                <xs:element
  ref="email2:LastVerbExecutionTime" minOccurs="0"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                                <xs:element ref="email2:ReceivedAsBcc"
  minOccurs="0"/>
                                                <xs:element ref="email2:Sender"
  minOccurs="0"/>
                                                <xs:element ref="email:Categories"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Timezone"
  minOccurs="0"/>
                                                <xs:element ref="calendar:AllDayEvent"
  minOccurs="0"/>
                                                <xs:element ref="calendar:BusyStatus"
  minOccurs="0"/>
                                                <xs:element ref="calendar:OrganizerName"
  minOccurs="0"/>
                                                <xs:element ref="calendar:OrganizerEmail"
  minOccurs="0"/>
                                                <xs:element ref="calendar:DtStamp"
  minOccurs="0"/>
                                                <xs:element ref="calendar:EndTime"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Location"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Reminder"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Sensitivity"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Subject"
  minOccurs="0"/>
                                                <xs:element ref="calendar:StartTime"
  minOccurs="0"/>
                                                <xs:element ref="calendar:UID"
  minOccurs="0"/>
                                                <xs:element ref="calendar:MeetingStatus"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Attendees"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Recurrence"
  minOccurs="0"/>
                                                <xs:element ref="calendar:Exceptions"
  minOccurs="0"/>
                                                <xs:element ref="calendar:ResponseRequested"
  minOccurs="0"/>
                                                <xs:element
  ref="calendar:AppointmentReplyTime" minOccurs="0"/>
                                                <xs:element ref="calendar:ResponseType"
  minOccurs="0"/>
                                                <xs:element
  ref="calendar:DisallowNewTimeProposal" minOccurs="0"/>
                                                <xs:element ref="contacts:Anniversary"
  minOccurs="0"/>
                                                <xs:element ref="contacts:AssistantName"
  minOccurs="0"/>
                                                <xs:element
  ref="contacts:AssistantPhoneNumber" minOccurs="0"/>
                                                <xs:element
  ref="contacts:AssistnamePhoneNumber" minOccurs="0"/>
                                                <xs:element ref="contacts:Birthday"
  minOccurs="0"/>
                                                <xs:element
  ref="contacts:Business2PhoneNumber" minOccurs="0"/>
                                                <xs:element
  ref="contacts:BusinessAddressCity" minOccurs="0"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                                <xs:element
ref="contacts:BusinessPhoneNumber" minOccurs="0"/>
                                                <xs:element ref="contacts:WebPage"
minOccurs="0"/>
                                                <xs:element
ref="contacts:BusinessAddressCountry" minOccurs="0"/>
                                                <xs:element ref="contacts:Department"
minOccurs="0"/>
                                                <xs:element ref="contacts:Email1Address"
minOccurs="0"/>
                                                <xs:element ref="contacts:Email2Address"
minOccurs="0"/>
                                                <xs:element ref="contacts:Email3Address"
minOccurs="0"/>
                                                <xs:element ref="contacts:BusinessFaxNumber"
minOccurs="0"/>
                                                <xs:element ref="contacts:FileAs"
minOccurs="0"/>
                                                <xs:element ref="contacts:Alias"
minOccurs="0"/>
                                                <xs:element ref="contacts:WeightedRank"
minOccurs="0"/>
                                                <xs:element ref="contacts:FirstName"
minOccurs="0"/>
                                                <xs:element ref="contacts:MiddleName"
minOccurs="0"/>
                                                <xs:element ref="contacts:HomeAddressCity"
minOccurs="0"/>
                                                <xs:element ref="contacts:HomeAddressCountry"
minOccurs="0"/>
                                                <xs:element ref="contacts:HomeFaxNumber"
minOccurs="0"/>
                                                <xs:element ref="contacts:HomePhoneNumber"
minOccurs="0"/>
                                                <xs:element ref="contacts:Home2PhoneNumber"
minOccurs="0"/>
                                                <xs:element
ref="contacts:HomeAddressPostalCode" minOccurs="0"/>
                                                <xs:element ref="contacts:HomeAddressState"
minOccurs="0"/>
                                                <xs:element ref="contacts:HomeAddressStreet"
minOccurs="0"/>
                                                <xs:element ref="contacts:MobilePhoneNumber"
minOccurs="0"/>
                                                <xs:element ref="contacts:Suffix"
minOccurs="0"/>
                                                <xs:element ref="contacts:CompanyName"
minOccurs="0"/>
                                                <xs:element ref="contacts:OtherAddressCity"
minOccurs="0"/>
                                                <xs:element
ref="contacts:OtherAddressCountry" minOccurs="0"/>
                                                <xs:element ref="contacts:CarPhoneNumber"
minOccurs="0"/>
                                                <xs:element
ref="contacts:OtherAddressPostalCode" minOccurs="0"/>
                                                <xs:element ref="contacts:OtherAddressState"
minOccurs="0"/>
                                                <xs:element ref="contacts:OtherAddressStreet"
minOccurs="0"/>
                                                <xs:element ref="contacts:PagerNumber"
minOccurs="0"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                                <xs:element ref="contacts:Title"
minOccurs="0"/>
                                                <xs:element
ref="contacts:BusinessAddressPostalCode" minOccurs="0"/>
                                                <xs:element ref="contacts:LastName"
minOccurs="0"/>
                                                <xs:element ref="contacts:Spouse"
minOccurs="0"/>
                                                <xs:element
ref="contacts:BusinessAddressState" minOccurs="0"/>
                                                <xs:element
ref="contacts:BusinessAddressStreet" minOccurs="0"/>
                                                <xs:element ref="contacts:JobTitle"
minOccurs="0"/>
                                                <xs:element ref="contacts:YomiFirstName"
minOccurs="0"/>
                                                <xs:element ref="contacts:YomiLastName"
minOccurs="0"/>
                                                <xs:element ref="contacts:YomiCompanyName"
minOccurs="0"/>
                                                <xs:element ref="contacts:OfficeLocation"
minOccurs="0"/>
                                                <xs:element ref="contacts:RadioPhoneNumber"
minOccurs="0"/>
                                                <xs:element ref="contacts:Picture"
minOccurs="0"/>
                                                <xs:element ref="contacts:Categories"
minOccurs="0"/>
                                                <xs:element ref="contacts:Children"
minOccurs="0"/>
                                                <xs:element ref="contacts2:CustomerId"
minOccurs="0"/>
                                                <xs:element ref="contacts2:GovernmentId"
minOccurs="0"/>
                                                <xs:element ref="contacts2:IMAddress"
minOccurs="0"/>
                                                <xs:element ref="contacts2:IMAddress2"
minOccurs="0"/>
                                                <xs:element ref="contacts2:IMAddress3"
minOccurs="0"/>
                                                <xs:element ref="contacts2:ManagerName"
minOccurs="0"/>
                                                <xs:element ref="contacts2:CompanyMainPhone"
minOccurs="0"/>
                                                <xs:element ref="contacts2:AccountName"
minOccurs="0"/>
                                                <xs:element ref="contacts2:NickName"
minOccurs="0"/>
                                                <xs:element ref="contacts2:MMS"
minOccurs="0"/>
                                                <xs:element ref="documentlibrary:LinkId"
minOccurs="0"/>
                                                <xs:element ref="documentlibrary:DisplayName"
minOccurs="0"/>
                                                <xs:element ref="documentlibrary:IsFolder"
minOccurs="0"/>
                                                <xs:element
ref="documentlibrary:CreationDate" minOccurs="0"/>
                                                <xs:element
ref="documentlibrary:LastModifiedDate" minOccurs="0"/>
                                                <xs:element ref="documentlibrary:IsHidden"
minOccurs="0"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                                <xs:element
ref="documentlibrary:ContentLength" minOccurs="0"/>
                                                <xs:element ref="documentlibrary:ContentType"
minOccurs="0"/>
                                                <xs:element ref="notes:Subject" minOccurs="0"
/>
                                                <xs:element ref="notes:MessageClass"
minOccurs="0"/>
                                                <xs:element ref="notes:LastModifiedDate"
minOccurs="0"/>
                                                <xs:element ref="notes:Categories"
minOccurs="0"/>
                                                <xs:element ref="tasks:Rtf" minOccurs="0"/>
                                                <xs:element ref="tasks:Subject" minOccurs="0"
/>
                                                <xs:element ref="tasks:Importance"
minOccurs="0" />
                                                <xs:element ref="tasks:UtcStartDate"
minOccurs="0"/>
                                                <xs:element ref="tasks:StartDate"
minOccurs="0"/>
                                                <xs:element ref="tasks:UtcDueDate"
minOccurs="0"/>
                                                <xs:element ref="tasks:DueDate"
minOccurs="0"/>
                                                <xs:element ref="tasks:Categories"
minOccurs="0"/>
                                                <xs:element ref="tasks:Recurrence"
minOccurs="0"/>
                                                <xs:element ref="tasks:Complete"
minOccurs="0"/>
                                                <xs:element ref="tasks:DateCompleted"
minOccurs="0"/>
                                                <xs:element ref="tasks:Sensitivity"
minOccurs="0" />
                                                <xs:element ref="tasks:ReminderTime"
minOccurs="0"/>
                                                <xs:element ref="tasks:ReminderSet"
minOccurs="0"/>
                                            </xs:all>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Range" type="xs:string" minOccurs="0"/>
                        <xs:element name="Total" type="xs:integer" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.14.2.1 Search

The <Search> element is the top-level element in the XML stream for the **Search** command. The element identifies the body of the HTTP **POST** as containing a **Search** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Store> (request only)<br><Status> (response only)<br><Response> (response only) | **Container** | 1...1 (required) |

##### 2.2.2.14.2.1.1 Status

The **Search** command response <Status> element indicates whether the server encountered an error while it was processing the search query.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Search> (response only) | None | **Integer** | 1 (required) |

The following table specifies valid values for the <Status> element as a child of the <Search> node in the search response.

| Value | Meaning |
|---|---|
| 1 | Success |
| 3 | Server error |

For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

The <Status> element value indicates only that the **Search** command was processed correctly. It does not indicate whether any matches were found. The <Total> and <Range> response XML elements indicate how many matches were found and returned, respectively.

The response will contain multiple <Status> elements. The <Status> element indicates the processing status of the overall **Search** command when the <Search> element is the immediate parent of the <Status> element. When the immediate parent of the <Status> element is the <Store> element (section 2.2.2.14.2.1.2.1.1), that <Status> element indicates the processing status for only that store. This structure was chosen to enable possible future expansion of the command to searching multiple locations, address lists, and contacts folders.

##### 2.2.2.14.2.1.2 Response

The <Response> element is a container for the results that are returned from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Search> (response only) | <Store> (response only) | **Container** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.14.2.1.2.1 Store

In the **Search** command request XML, the <Store> element is a container for elements that specify the location, string, and options for the search. In the **Search** command response XML, the <Store> element contains the <Status>, <Result>, <Range>, and <Total> elements that contain the returned mailbox entries.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Search> (request)<br><Response> (response) | <Name> (request)<br><Query> (request only)<br><Options> (request only)<br><Status> (response only)<br><Result> (response only)<br><Range> (response only)<br><Total> (response only) | **Container** | 1...1 (required) |

#### 2.2.2.14.2.1.2.1.1 Status

The <Status> element indicates whether the server encountered an error while it was processing the **Search** query for the specified store.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (response only) | None | **Integer** | 1 (required) |

The following table specifies valid values for the <Status> element as a child of the <Store> element in the **Search** response.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | The request was invalid. | One or more of the client's search parameters was invalid. | Item | If the user formatted the request, prompt the user to retry with different options. |
| 3 | An error occurred on the server. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry. |
| 4 | Bad link. | A bad link was supplied. | Global | Prompt user to reformat link. |
| 5 | Access denied. | Access was denied to the resource | Global | Prompt the user. |
| 6 | Not found. | Resource was not found. | Global | Prompt the user. |
| 7 | Connection failed. | Failed to connect to the resource. | Global | Prompt the user. Sometimes these are transient, so retry. If it continues to fail, point user to administrator. |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 8 | Too complex. | The query was too complex. | Global | Reduce the complexity of the query. Prompt user if necessary. |
| 10 | Timed out. | The search timed out. | Global | The search timed out. Retry with or without rebuilding results. If it continues, contact the Administrator. |
| 11 | **FolderSync** required. | The folder hierarchy is out of date. | Global | Issue a **FolderSync** and try again. |
| 12 | End of retrievable range warning. | The requested range has gone past the end of the range of retrievable results. | Local | Prompt the user that there are no more results that can be fetched, and the user might consider restricting their search query. |
| 13 | Access blocked. | Access is blocked to the specified **resource**. | Global | Prompt the user. |
| 14 | Credentials required. | To complete this request, basic credentials are required. | Global | If over a trusted connection, supply the basic credentials from the user (prompt if necessary). Otherwise fail as if the access denied status code was provided. |

For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.14.2.1.2.1.2 Result

The **Search** command response <Result> element is a container for an individual matching mailbox item.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (response only) | <airsync:Class> (response only)<br><LongId> (response only)<br><airsync:CollectionId> (response only)<br><Properties> (response only) | **Container** | 1...N (required) |

One <Result> element is present for each match that is found. If no matches are found, an empty <Result> element is present in the <Store> container element of the response XML.

Inside the <Result> element, the <Properties> element contains a list of nonempty text properties on the entry.

When the store being searched is the mailbox:

- There is one <Result> element for each match that is found in the mailbox. If no matches are found, an empty <Result> element is present in the <Store> container element of the response XML.
- Inside the <Result> element, the <Properties> element contains a list of requested properties for the mailbox item.



Copyright © 2010 Microsoft Corporation.

When the store that is being searched is the document library:

- The first result that is returned in the **Search** response is the metadata for the root folder or item to which the <documentlibrary:LinkId> is pointing. The client can choose to ignore this entry if it does not require it.
- If the <documentlibrary:LinkId> in the request points to a folder, the metadata properties of the folder are returned as the first item, and the contents of the folder are returned as subsequent results. The <Range> element applies to these results with no difference; for example, the index 0 would always be for the root item to which the link is pointing.
- If the <documentlibrary:LinkId> in the request points to an item, only one result is returned: the metadata for the item.
- Inside the <Result> element, the <Properties> element contains a list of requested properties for the mailbox item.

#### 2.2.2.14.2.1.2.1.2.1 airsync:Class

The <airsync:Class> element specifies the classes that the client wants returned for a given collection. It is defined as an element in the AirSync namespace.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only)<br><Schema> (request only)<br><Result> (response only) | None | String | 1…1 (required, 1 per <Result> element) |

The valid <airsync:Class> element values are:

- Tasks
- Email
- Calendar
- Contacts
- Notes
- SMS<41>

The **Search** request can include one or more <airsync:Class> elements in the request to limit the type of data included in the **Search** response.

If the <airsync:Class> element is included as a child of any element other than the <And> element, the server responds with a <Status> value of 8 (SearchTooComplex).

#### 2.2.2.14.2.1.2.1.2.2 LongId

The <LongId> element specifies a unique identifier that is assigned by the server to each result set that is being returned in the **Search** response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Result> (response only) | None | **String** (up to 256 characters) | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The value of the <LongId> element can be used in the <LongId> parameter of the **ItemOperations**, **SmartReply**, **SmartForward**, or **MeetingResponse** command requests to reference the result set.

The client MUST store the value of <LongId> as an opaque string of up to 256 characters.

#### 2.2.2.14.2.1.2.1.2.3 airsync:CollectionId

The <airsync:CollectionId> element specifies the folder in which the item was found.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <And> (request only)<br><Result> (response only) | None | **String** | 0...N (optional) |

#### 2.2.2.14.2.1.2.1.2.4 Properties

In a **Search** response, the <Properties> element encapsulates the properties for each search result.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Result> (response only) | <airsyncbase:Body> as specified in [MS-ASAIRS] section 2.2.2.4<br><airsyncbase:BodyPart> as specified in [MS-ASAIRS] section 2.2.2.5<br><gal:Picture><br>Data elements from the content classes. For more details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASNOTE], and [MS-ASTASK]. | **Container** | 1...1 (Required, response only) |

The **Search** command response <Properties> element is a container for properties that apply to an individual entry that matches the <Query> element search string. For example, the <Properties> element contains an element for each nonempty, text-valued GAL property that is attached to the matching GAL entry. Only those properties that are attached to the specific GAL entry are returned; therefore different sets of properties can be returned in the response XML for different matching GAL entries.

Each element in the <Properties> container is scoped to the appropriate namespace that is specified in the top-level **Search** element.

#### 2.2.2.14.2.1.2.1.2.4.1 gal:Picture

The <gal:Picture> element<42> contains the data related to the contact photos.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Properties> (response only) | <MaxSize> (request only)<br><MaxPictures> (request only)<br><Status> (response only) | **Container** (request or response)<br>**Empty** (request only) | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <gal:Data> (response only) | | |

#### 2.2.2.14.2.1.2.1.2.4.1.1 Status

The <Status> element indicates whether the server encountered an error while it was processing the **Search** query for the contact photos.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> (response only)<br><Search> (response only)<br><gal:Picture> (response only) | None | **Integer** | 1 (required) |

The following table shows valid values for the <Status> element when it is returned as a child of the <gal:Picture> element.<43>

| Value | Meaning |
|---|---|
| 1 | The contact photo was retrieved successfully. |
| 173 | The user does not have a contact photo. |
| 174 | The contact photo exceeded the size limit set by the <MaxSize> element (section 2.2.2.14.1.1.1.3.7.1). |
| 175 | The number of contact photos returned exceeded the size limit set by the <MaxPictures> element (section 2.2.2.14.1.1.1.3.7.2). |

For information about <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.14.2.1.2.1.2.4.1.2 gal:Data

The <gal:Data> element<44> contains the binary data of the contact photo.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <gal:Picture> (response only) | None | **Byte array** | 0…1 (optional) |

#### 2.2.2.14.2.1.2.1.3 Range

The **Search** command <Range> element is used in both the request and response XML documents. In the request XML, the <Range> element specifies the range of matching entries to return. In the response XML, the <Range> element specifies the number of matching entries that are being returned.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only)<br><Store> (response only) | None | Zero-based range in the form m-n | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The <Range> element value specifies a number of entries, but indicates different things depending on whether the element is in the request or the response XML.

The format of the <Range> element value is in the form of a zero-based index specifier, formed with a numeric value, a hyphen, and another numeric value: "m-n." The m and n indicates the lowest and highest index of a zero-based array that would hold the items. For example, a <Range> element value of 0–9 indicates 10 items, and 0–10 indicates 11 items. A <Range> element value of 0–0 indicates 1 item.

If the request does not include a <Range> element, the default <Range> value for each <Store> type is used. The following table identifies the default <Range> values and maximum results returned for each <Store> type:

| Store value | Default Range value | Maximum results returned |
|---|---|---|
| Mailbox | 0-99 | 100 |
| DocumentLibrary | 0-999 | 1000 |
| GAL | 0-99 | 100 |

If the <Range> value specified in the request exceeds the default range value, a <Status> value of 12 is returned to indicate that the maximum range has been exceeded, as specified in section 2.2.2.14.2.1.2.1.1.

In the request XML, the <Range> element value specifies the range of entries to be returned to the client.

In the response XML, the <Range> element value specifies the actual number of entries that are returned in the response. The <Total> element in the response XML provides an estimate of the total number of entries that matched the <Query> element value.

Search results are stored in a search folder on the server. This way, when a client comes back with the same query but a new row range, rows are pulled from the result set that is currently stored in the search folder. The entire result set does not have to be rebuilt.

#### 2.2.2.14.2.1.2.1.4 Total

The **Search** command response XML element <Total> provides an estimate of the total number of mailbox entries that matched the search <Query> element value.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Store> | None | **Integer** | 1...1 (required) |

The value of the <Total> element does not always equal the number of entries that are returned. To determine the number of entries that are returned by the **Search** command, use the <Range> element value.

The <Total> element indicates the number of entries that are available. In cases where all the results are returned in the response XML, the value of the <Total> element is one more than the end-index value that is provided in the <Range> element. For example, if the **Search** command returns 15 entries, the value of the <Range> element is 0–14, while the value of the <Total> element is 15.



Copyright © 2010 Microsoft Corporation.

The <Total> element is used by clients to determine whether more matching entries were found in the mailbox than have been returned by the **Search** command. For example, a client might perform an initial search and specify a requested <Range> of 0–4 (return 5 entries maximum). If the <Total> element indicates that there are actually 25 matching items, the client can then enable the user to retrieve the full results.

### 2.2.2.15 SendMail

The **SendMail** command is used by clients to send MIME-formatted e-mail messages to the server.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, clients SHOULD use the <SaveInSentItems> element to automatically have the messages saved on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command. Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

Note that the From: MIME header in the outgoing message is set on the server to the primary e-mail address of the authenticated user.

The ComposeMail namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ComposeMail namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.15.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **SendMail** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:rm="RightsManagement:">

    <xs:import namespace="RightsManagement:"/>

    <xs:complexType name="EmptyTag" />
    <xs:element name="SendMail">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
                            <xs:maxLength value="40"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="AccountId" type="xs:string" minOccurs="0"/>
                <xs:element name="SaveInSentItems" type="tns:EmptyTag" minOccurs="0"/>
                <xs:element name="Mime" type="xs:string"/>
                <xs:element ref="rm:TemplateID" minOccurs="0"/>
            </xs:all>
        </xs:complexType>
```



Copyright © 2010 Microsoft Corporation.

```
    </xs:element>
</xs:schema>
```

#### 2.2.2.15.1.1 SendMail

The <SendMail> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SendMail** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS-ASRM] section 2.2.2.2<br><Status> (response only) | **Container** | 1...1 (required) |

##### 2.2.2.15.1.1.1 ClientId

The <ClientId> element specifies the client's unique message ID (MID).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> (request only) | None | **String** (Up to 40 characters) | 1...1 (required) |

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.

##### 2.2.2.15.1.1.2 AccountId

The <AccountId> element<45> identifies the account from which an e-mail is sent.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> | None | **String** | 0...1 (optional) |

If the <AccountId> element is not present in the **SendMail** request, the server sends the e-mail using the account identified by the <PrimarySmtpAddress> element returned in the **Settings** command response.

If <AccountId> is included in the request, the value MUST equal one of the <AccountId> values included in the **Settings** command response (section 2.2.2.16.2).

The server MUST validate that the <AccountId> value provided is valid for sending e-mail. A <Status> value of 166 is returned if the <AccountId> value is not valid. A <Status> value of 167 is returned if the account does not support sending e-mail.

**Note** The server sends the e-mail using the <AccountId> value specified by the <AccountId> element, and not the account specified by the <From> element.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.15.1.1.3 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should be stored in the Sent Items folder. If the <SaveInSentItems> element is present, the message is stored -- if not present, the message is not stored.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> (request only) | None | **Empty tag** | 0...1 (optional) |

The <SaveInSentItems> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <SaveInSentItems/> tag.

#### 2.2.2.15.1.1.4 Mime

The <Mime> element contains the MIME-encoded message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> (request only) | None | **Byte array** | 1...1 (required) |

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

If the message contains a meeting request, the <Mime> element contains the details of meeting in iCalendar format [MS-OXCICAL] or **Transport Neutral Encapsulation Format (TNEF)** format [MS-OXTNEF]. As specified in [RFC2447] section 3.4, iCalendar meeting requests have a content type of text/calendar with the *method* parameter set to **REQUEST**.

### 2.2.2.15.2 Response

The following code shows the **SendMail** command response XSD [XMLSCHEMA1].

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="ComposeMail:"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="ComposeMail:"
   elementFormDefault="unqualified">
   <xs:element name="SendMail">
      <xs:complexType>
         <xs:all>
            <xs:element name="Status" type="xs:integer" minOccurs="0"/>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

If the message was sent successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.15.2.1 SendMail

The <SendMail> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SendMail** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS-ASRM] section 2.2.2.2<br><Status> (response only) | **Container** | 1...1 (required) |

##### 2.2.2.15.2.1.1 Status

The <Status> element indicates the reason for the failure of a **SendMail** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> (response only) | None | **Integer** | 0...1 (optional) |

If the command succeeds, no XML body is returned in the response, as specified in section 2.2.2.15.2. If the command fails, the <Status> element contains a code that indicates the type of failure.

Valid <Status> values are listed in 2.2.3.

### 2.2.2.16 Settings

The **Settings** command supports get and set operations on global properties and Out of Office (OOF) settings for the user. The **Settings** command also sends device information to the server for display in the user and IT interfaces, implements the device password/personal identification number (PIN) recovery, and retrieves a list of the user's e-mail addresses.

The <Get> and <Set> operations act on **named properties**. In the context of the <Get> and <Set> operations, each named property can contain a set of property-specific data nodes.

The **Settings** command can contain multiple <Get> and <Set> requests and responses in any order. The implication of this batching mechanism is that commands are executed in the order in which they are received and that the ordering of <Get> and <Set> responses will match the order of those commands in the request.

The following is the generic form of the **Settings** request:

```
<Settings>
    <PropertyName>
        Data nodes
    </PropertyName>
    ...
</Settings>
```



Copyright © 2010 Microsoft Corporation.

The <PropertyName> is a named property (that is, the actual name of the property). The **Settings** command can be used on the following named properties:

- <OOF>
- <DevicePassword>
- <DeviceInformation>
- <UserInformation>

Clients SHOULD send <DeviceInformation> parameters in a <Set> block to the server as soon as the client has been provisioned, and before the **FolderSync** command, so that the server can use this information to determine what the device has access to.<46>

The argument or data nodes are <Get> or <Set>, which can also have their own arguments. It is up to the individual property handlers to parse and interpret them as necessary.

It is possible to have between zero and four <PropertyName> nodes in a **Settings** request (each of the four named properties can appear zero or one time in a request). Each property MUST be processed in order. There can be cases in which one property call affects another property call or the same property is in the **Settings** request more than once. The responses will come back in the same order in which they were requested.

Each response message contains a <Status> value for the command, which addresses the success or failure of the **Settings** command, followed by <Status> values for each of the changes made to the <Oof>, <DeviceInformation>, <DevicePassword> or <UserInformation> elements.

The <Status> node MUST return Success if **Settings** is returning property responses. If the command was not successful, the processing of the request cannot begin, no property responses are returned, and <Status> MUST indicate a protocol error.

Any error other than a protocol error is returned in the status codes of the individual property responses. All property responses, regardless of the property, MUST contain a status element to indicate success or failure. This status node MUST be the first node in the property response.

The Settings namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the Settings namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.16.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **Settings** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Settings:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Settings:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:simpleType name="NonEmptyStringType">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
    </xs:restriction>
</xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
<xs:complexType name="EmptyTag" />

<xs:simpleType name="DeviceInformationStringType">
    <xs:restriction base="xs:string">
        <xs:maxLength value="1024"/>
    </xs:restriction>
</xs:simpleType>


    <xs:element name="DeviceInformation">
        <xs:complexType>
            <xs:choice>
                <xs:element name="Set">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="Model" type="tns:DeviceInformationStringType"
minOccurs="0"/>
                            <xs:element name="IMEI" type="tns:DeviceInformationStringType"
minOccurs="0"/>
                            <xs:element name="FriendlyName"
type="tns:DeviceInformationStringType" minOccurs="0"/>
                            <xs:element name="OS" type="tns:DeviceInformationStringType"
minOccurs="0"/>
                            <xs:element name="OSLanguage"
type="tns:DeviceInformationStringType" minOccurs="0"/>
                            <xs:element name="PhoneNumber"
type="tns:DeviceInformationStringType" minOccurs="0"/>
                            <xs:element name="UserAgent"
type="tns:DeviceInformationStringType" minOccurs="0"/>
                            <xs:element name="EnableOutboundSMS" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:integer">
                                        <xs:minInclusive value="0"/>
                                        <xs:maxInclusive value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="MobileOperator"
type="tns:DeviceInformationStringType" minOccurs="0"/>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="Settings">
        <xs:complexType>
            <xs:all>
                <xs:element name="RightsManagementInformation" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Get" type="tns:EmptyTag"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Oof" minOccurs="0">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element name="Get" minOccurs="0">
```



Copyright © 2010 Microsoft Corporation.

```
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="BodyType"
type="tns:NonEmptyStringType"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Set">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="OofState"
type="tns:NonEmptyStringType" minOccurs="0"/>
                                        <xs:element name="StartTime"
type="tns:NonEmptyStringType" minOccurs="0"/>
                                        <xs:element name="EndTime"
type="tns:NonEmptyStringType" minOccurs="0"/>
                                        <xs:element name="OofMessage" minOccurs="0"
maxOccurs="3">
                                            <xs:complexType>
                                                <xs:all>
                                                    <xs:element name="AppliesToInternal"
type="tns:EmptyTag" minOccurs="0"/>
                                                    <xs:element name="AppliesToExternalKnown"
type="tns:EmptyTag" minOccurs="0"/>
                                                    <xs:element
name="AppliesToExternalUnknown" type="tns:EmptyTag" minOccurs="0"/>
                                                    <xs:element name="Enabled"
type="tns:NonEmptyStringType" minOccurs="0"/>
                                                    <xs:element name="ReplyMessage"
type="xs:string" minOccurs="0"/>
                                                    <xs:element name="BodyType"
type="tns:NonEmptyStringType" minOccurs="0"/>
                                                </xs:all>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
                <xs:element name="DevicePassword" minOccurs="0">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element name="Set">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Password" >
                                           <xs:simpleType>
                                              <xs:restriction base="xs:string">
                                                 <xs:maxLength value="255"/>
                                              </xs:restriction>
                                           </xs:simpleType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
```



Copyright © 2010 Microsoft Corporation.

```
                <xs:element ref="tns:DeviceInformation" minOccurs="0"/>
                <xs:element name="UserInformation" minOccurs="0">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element name="Get" type="tns:EmptyTag"/>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

#### 2.2.2.16.1.1 Settings

The <Settings> element is the top-level element in the XML stream for the **Settings** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Oof><br><DeviceInformation><br><DevicePassword><br><UserInformation><br><Status> (response only) | **Container** | 1...1 (required) |

The <Settings> element encapsulates one or more named property nodes that contain actions and arguments that apply to those properties.

##### 2.2.2.16.1.1.1 RightsManagementInformation

The <RightsManagementInformation> element<47> is a container node that is used to request rights management information settings.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> (request and response) | <Get> (request or response)<br><Status> (response only) | **Container** | 0…1 (optional) |

If the <RightsManagementInformation> element is specified in the request, it MUST include the <Get> element.

###### 2.2.2.16.1.1.1.1 Get

The <Get> element enables the client to retrieve <RightsManagementInformation> settings from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <RightsManagementInformation> | None (request only)<br><rm:RightsManagementTemplates> as specified in [MS-ASRM] section 2.2.2.4 | **Empty tag** (request only)<br>**Container** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | (response only) | (response only) | |

#### 2.2.2.16.1.1.2 Oof

The <Oof> element specifies a named property node for retrieving and setting out of office (OOF) information.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Get><br><Set> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |

The **Settings** command supports <Get> and <Set> operations for the <Oof> element. The <Oof> element enables a user to do the following:

- Specify whether the user is currently out of office.
- Schedule an out of office message to be sent between a particular start date and end date.
- Specify the message that is to be shown to various audiences when the mobile device user is out of office.

<Oof> Get Request and Response

The <Get> element within the <Oof> element enables the client to retrieve OOF information from the server. The client specifies the <BodyType> to be retrieved and the server will return all OOF information and messages.

There is one <OofMessage> node per audience in an <Oof> <Get> response. If the sender group is allowed, but is disabled and has no reply message (specified by the <ReplyMessage> element), an <OofMessage> node is still reported to the client.

If the client does not receive a group, it is presumably because the client does not have permission to enter settings for that group; in such a case, any attempt to set those properties results in an Access Denied status code.

<Oof> Set Request and Response

The <Set> element enables the client to set the OOF status, time OOF, and OOF messages for one or more of the following groups:

- Internal
- External Known Senders (such as contacts)
- External Unknown Senders

#### 2.2.2.16.1.1.2.1 Get

The <Get> element enables the client to retrieve <Oof> settings from the server.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Oof> | <BodyType> | **Container** | 0…1 (optional) |

In an <Oof> request, the client specifies the <BodyType> to be retrieved and the server will return all OOF settings and messages for that body type.

#### 2.2.2.16.1.1.2.1.1 BodyType

The <BodyType> element is a string that specifies the format of the OOF message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (request only) | None | **String** (not NULL) | 1 (required) |

The following are the permitted values for the <BodyType> element:

- *Text*
- *HTML*

If <BodyType> has the value HTML, all message strings are sent in the HTML format. If <BodyType> has the value *Text*, the message strings are sent in **plain text**. Because there is no default value, the <BodyType> node MUST be present on all <Get> operations and on any <OofMessage> where the <ReplyMessage> has been set.

### 2.2.2.16.1.1.2.2 Set

The <Set> element enables the client to set <Oof> information on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Oof> (request only) | <OofState><br><StartTime><br><EndTime><br><OofMessage> | **Container** | 0...1 (optional) |

When setting the <OOF> element, the client can set the following:

- OOF state
- Start time and end time, if the user wants to schedule an OOF message.
- OOF message or messages for one or more of the supported audiences

#### 2.2.2.16.1.1.2.2.1 OofState

The <OofState> element specifies the availability of the OOF property.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (OOF response only)<br><Set> (OOF request only) | None | **Integer** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The following table lists the valid values for <OofState>.

| Value | Meaning |
|---|---|
| 0 | The <OOF> property is disabled. |
| 1 | The <OOF> property is global. |
| 2 | The <OOF> property is time-based. |

#### 2.2.2.16.1.1.2.2.2 StartTime

The <StartTime> element is used with the <EndTime> element to specify the range of time during which the user is out of office.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | None | **DateTime** | 0...1 (optional) |

The <StartTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

In a <Set> <Oof> node, both <StartTime> and <EndTime> MUST be included in the **Settings** request, or neither <StartTime> or <EndTime> MUST be included in the **Settings** request. If either <StartTime> or <EndTime> is included in the request without the other, a <Status> value of 2 is returned as a child of the <Oof> element.

#### 2.2.2.16.1.1.2.2.3 EndTime

The <EndTime> element is used with the <StartTime> element to specify the range of time during which the user is out of office.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | None | **DateTime** | 0...1 (optional) |

The <EndTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property or within the <Set> element of the **Settings** request for the <Oof> property.

In a <Set> <Oof> node, both <StartTime> and <EndTime> MUST be included in the **Settings** request, or neither <StartTime> or <EndTime> MUST be included in the **Settings** request. If either <StartTime> or <EndTime> is included in the request without the other, a <Status> value of 2 is returned as a child of the <Oof> element.

#### 2.2.2.16.1.1.2.2.4 OofMessage

The <OofMessage> element contains a set of elements that specify the OOF message for a particular audience.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only) | <AppliesToInternal> | **Container** | 0...3 |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<Oof> request only) | <AppliesToExternalKnown><br><AppliesToExternalUnknown><br><Enabled><br><ReplyMessage><br><BodyType> | | |

The <Oof> property supports the following three audiences for an OOF message:<48>

- Internal—A user who is in the same organization as the sending user.
- Known external—A user who is outside the sending user's organization, but is represented in the sending user's contacts.
- Unknown external—A user who is outside the sending user's organization and is not represented in the sending user's contacts.

The presence of one of the following elements, which are mutually exclusive, indicates the audience to which an <OOF> message pertains:

- <AppliesToInternal>—The OOF message is relevant to an internal audience.
- <AppliesToExternalKnown>—The OOF message is relevant to a known external audience.
- <AppliesToExternalUnknown>—The OOF message is relevant to an unknown external audience.

There is one <OofMessage> node per audience in an OOF <Get> response. If a sender group is allowed, but is disabled and has no reply message (specified by the <ReplyMessage> element), an <OofMessage> node is reported to the client. If <AppliesToExternalKnown> or <AppliesToExternalUnknown> are not allowed and are disabled by the administrator but are sent by the client in the <Set> request, <Set> returns a successful <Status> value of 1 even though the user does not have access to these settings. Similarly, <AppliesToExternalKnown> and <AppliesToExternalUnknown> are returned to the client in a <Get> response even if the sender group is not allowed and is disabled.

In an OOF <Set> request, the client MUST NOT include the same AppliesTo* element in more than one <OofMessage> element.

#### 2.2.2.16.1.1.2.2.4.1 AppliesToInternal

The <AppliesToInternal> element indicates that the OOF message applies to internal users. (An internal user is a user who is in the same organization as the sending user.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |

The <AppliesToInternal> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToInternal/> tag.



Copyright © 2010 Microsoft Corporation.

When the <AppliesToInternal> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to internal users.

The following are the peer elements of the <AppliesToInternal> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF message itself.
- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.1.1.2.2.4.2 AppliesToExternalKnown

The <AppliesToExternalKnown> element indicates that the OOF message applies to known external users. (A known external user is a user who is outside the sending user's organization, but is represented in the sending user's contacts.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |

The <AppliesToExternalKnown> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalKnown/> tag.

When the <AppliesToExternalKnown> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to known external users.

The following are the peer elements of the <AppliesToExternalKnown> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.1.1.2.2.4.3 AppliesToExternalUnknown

The <AppliesToExternalUnknown> element indicates that the OOF message applies to unknown external users. (An unknown external user is a user who is outside the sending user's organization and is not represented in the sending user's contacts.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |



Copyright © 2010 Microsoft Corporation.

The <AppliesToExternalUnknown> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalUnknown/> tag.

When the <AppliesToExternalUnknown> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to unknown external users.

The following are the peer elements of the <AppliesToExternalUnknown> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.1.1.2.2.4.4 Enabled

The <Enabled> element specifies whether an OOF message is sent to this audience while the sending user is OOF.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **String** | 0...1 (optional) |

The <Enabled> element is used in the OOF <Get> response to retrieve the current value. The <Enabled> element is used in the OOF <Set> request to set the value.

The value of <Enabled> is 1 if an OOF message is sent while the sending user is OOF; otherwise, the value is 0.

#### 2.2.2.16.1.1.2.2.4.5 ReplyMessage

The <ReplyMessage> element specifies the message to be shown to a particular audience when the user is OOF.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **String** | 0...1 (optional) |

The <ReplyMessage> can be used in an OOF <Get> response to convey the requested OOF message, or in an OOF <Set> request to set the message that the client wants to send to a particular audience. In a <Set>, any <ReplyMessage> MUST also specify a <BodyType>.

The <OOF> property supports the following three audiences for an OOF message:

- Internal—A user who is in the same organization as the sending user.
- Known external—A user who is outside the sending user's organization, but is represented in the sending user's contacts.
- Unknown external—A user who is outside the sending user's organization and is not represented in the sending user's contacts.



Copyright © 2010 Microsoft Corporation.

The presence of one of the following elements, which are mutually exclusive, indicates the audience to which an OOF message pertains:

- <AppliesToInternal>—The OOF message is relevant to an internal audience.
- <AppliesToExternalKnown>—The OOF message is relevant to a known external audience.
- <AppliesToExternalUnknown>—The OOF message is relevant to an unknown external audience.

#### 2.2.2.16.1.1.2.2.4.6 BodyType

The <BodyType> element is a string that specifies the format of the OOF message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **String** (not NULL) | 1 (required) if a message is set; otherwise, 0…1 (optional) |

The following are the permitted values for the <BodyType> element:

- *Text*
- *HTML*

If <BodyType> has the value **HTML**, all message strings are sent in the HTML format. If <BodyType> has the value *Text*, the message strings are sent in plain text. Because there is no default value, the <BodyType> node MUST be present on all <Get> operations and on any <OofMessage> where the <ReplyMessage> has been set.

### 2.2.2.16.1.1.3 DevicePassword

The <DevicePassword> element is a container node that is used to send the recovery password of the client device to the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Set> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |

Use the <Set> operation on the <DevicePassword> property to enable the device to send or store a recovery password on the server. The recovery password is stored in the user's mailbox and can be retrieved by the administrator or the end-user if the user forgets his or her password.

#### 2.2.2.16.1.1.3.1 Set

The <Set> element enables the client to set <DevicePassword> settings on the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <DevicePassword> (request only) | <Password> (request only) | **Container** | 0...1 (optional) |

Setting the <DevicePassword> enables the client to set or clear the recovery password of the device.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.16.1.1.3.1.1 Password

The <Password> element specifies the recovery password of the client device, which is stored by the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DevicePassword> request only) | None | **String** | 1...1 (required) |

The value of the <Password> element has a maximum length of 255 characters.

To clear an existing recovery password, the client MUST send a <Set> request with an empty <Password> element.

### 2.2.2.16.1.1.4 DeviceInformation

The <DeviceInformation> element is the container node that is used for sending the client device's properties to the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Set> (request only)<br><Status> (response only) | <Container> | 0...1 (optional) |

Clients SHOULD use the **Provision** command to send <DeviceInformation> to the server.<49>

When the **Settings** command is used to send the <DeviceInformation> element, it sends the following information about a client device to the server:

- Device model
- International Mobile Equipment Identity (IMEI)
- Device friendly name
- Device operating system
- Telephone number
- Device operating system language
- User Agent
- Whether to enable outbound SMS (see [MS-ASMS])
- Mobile operator name

The device information is represented as a list of settings under the <DeviceInformation> node in the **Settings** command. <DeviceInformation> has only one child element, <Set>, which contains the list of device information items in the request and the status in the response. The <DeviceInformation> element supports only the <Set> elements because this information is write-only from the device.

#### 2.2.2.16.1.1.4.1 Set

The <Set> element enables the client to set <DeviceInformation> settings on the server.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <DeviceInformation> (request only) | <Model><br><IMEI><br><FriendlyName><br><OS><br><OSLanguage><br><PhoneNumber><br><UserAgent><br><EnableOutboundSMS><br><MobileOperator> | **Container** | 0...1 (optional) |

Clients SHOULD send <DeviceInformation> parameters to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command, so that the server can use this information to determine what the device has access to.<50>

<Set> enables the client to specify values for any of the <DeviceInformation> parameters. The following statements apply to the <Set> element request implementation:

- The client SHOULD specify all supported <DeviceInformation> parameters in the <Set> request. An error is not returned if all <DeviceInformation> parameters are not set.
- The client or server makes no assumptions about ordering. The <DeviceInformation> parameters can be specified in any order.
- To delete a given <DeviceInformation> value, the client MUST send the <Set> element as an empty element.

#### 2.2.2.16.1.1.4.1.1 Model

The <Model> element specifies a name that generally describes the device of the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

The descriptive name of the device can be any string that the client chooses, typically a general description of the device. For example, the name of the manufacturer, the model name, or the model number can be used. The server does not perform any validation of this string, so the client can submit any string.

#### 2.2.2.16.1.1.4.1.2 IMEI

The <IMEI> element specifies a 15-character code that MUST uniquely identify a device.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The server does not validate the IMEI format.

The device ID parameter that is currently included in the request URL is not precisely defined; protocol implementers are free to populate the field as they want. To enable workable inventory-type report generation, an ID that uniquely identifies a device in the space of all devices is required. The <IMEI> element satisfies this requirement.

#### 2.2.2.16.1.1.4.1.3 FriendlyName

The <FriendlyName> element specifies a name that MUST uniquely describe the client device.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

The friendly name of the device is a string that is meaningful to the user. The server does not validate this value.

The friendly name of the device is specified during partnership creation if the user creates a desktop-device partnership with a desktop. For more information about creating a desktop-device partnership, see [MSDN-ADDP].

#### 2.2.2.16.1.1.4.1.4 OS

The <OS> element specifies the operating system of the client device.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

Some information about the operating system of the device can be collected from the user agent string that is associated with requests from that client. The mapping from user agent to operating system is not one to one, however, and therefore does not provide sufficient information to troubleshoot and establish an inventory.

The <OS> element is a string value that enables the client to precisely specify the operating system of the device. The server does not perform any validation of this value, but clients SHOULD use the following convention:

<Operating System Product Name> <Operating System Major Version> <Operating System Minor Version>

#### 2.2.2.16.1.1.4.1.5 OSLanguage

The <OSLanguage> element specifies the language that is used by the operating system of the client device.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request | None | **String** (up to 1024 | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| only) | | characters) | |

Knowledge of the user's language facilitates localization if the server is required to send localizable content to the client device. The server does not validate the value of the <OSLanguage> element.

#### 2.2.2.16.1.1.4.1.6 PhoneNumber

The <PhoneNumber> element specifies a unique number that identifies the client device.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

The telephone number facilitates troubleshooting and device management by providing a well-known and unique identifier for the client device. The server does not validate the value of the <PhoneNumber> element.<51>

#### 2.2.2.16.1.1.4.1.7 UserAgent

The <UserAgent> element specifies the user agent.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

The <UserAgent> element SHOULD contain the information in the User-Agent header. The User-Agent header SHOULD be removed from the HTTP request. The server does not validate the value of the <UserAgent> element.

#### 2.2.2.16.1.1.4.1.8 EnableOutboundSMS

The <EnableOutboundSMS> element specifies whether the server will send outbound SMS messages through the mobile device. For more details, see [MS-ASMS].<52>

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Set> (<DeviceInformation> request only) | None | **Integer** | 0...1 (optional) |

If this element is set to 1, then the mobile device can be used to send outbound SMS messages composed on the server; otherwise, the mobile device cannot be used to send such outbound SMS messages.<53>

#### 2.2.2.16.1.1.4.1.9 MobileOperator

The <MobileOperator> element specifies the name of the mobile operator to which a mobile device is connected. For more details, see [MS-ASMS].<54>



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| Set (<DeviceInformation> request only) | None | **String** (up to 1024 characters) | 0...1 (optional) |

### 2.2.2.16.1.1.5 UserInformation

The <UserInformation> element is a container node that is used to request a list of a user's e-mail addresses from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Get><br><Status> (<UserInformation> response only) | **Container** | 0...1 (optional) |

The list of a user's e-mail addresses can be useful, for example, for ensuring that the user is not included when performing a Reply to All operation to an e-mail message.

In a request, the <UserInformation> element contains the <Get> element to indicate that the server is to return all available e-mail addresses for the user.

The **Settings** command supports read-only access to the list of a user's various e-mail addresses via the <Get> element. The client is unable to write this information.

#### 2.2.2.16.1.1.5.1 Get

The <Get> element enables the client to retrieve <UserInformation> settings from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <UserInformation> | None (request only)<br><EmailAddresses> (response only)<br><Accounts> (response only) | **Empty tag** (request only)<br>**Container** (response only) | 1 (required) |

### 2.2.2.16.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **Settings** command response.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="Settings:" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:rm="RightsManagement:" targetNamespace="Settings:" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:import namespace="RightsManagement:"/>
  <xs:element name="Settings">
    <xs:complexType>
      <xs:all>
        <xs:element name="Status" type="xs:integer" minOccurs="0"/>
        <xs:element name="Oof" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="Status" type="xs:integer" minOccurs="0"/>
```



Copyright © 2010 Microsoft Corporation.

```
        <xs:element name="Get" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="OofState" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:enumeration value="0"/>
                    <xs:enumeration value="1"/>
                    <xs:enumeration value="2"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="StartTime" type="xs:dateTime"/>
              <xs:element name="EndTime" type="xs:dateTime"/>
              <xs:element name="OofMessage" minOccurs="0" maxOccurs="3">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="AppliesToInternal" minOccurs="0"/>
                    <xs:element name="AppliesToExternalKnown" minOccurs="0"/>
                    <xs:element name="AppliesToExternalUnknown" minOccurs="0"/>
                    <xs:element name="Enabled" type="xs:integer" minOccurs="0"/>
                    <xs:element name="ReplyMessage" type="xs:string" minOccurs="0"/>
                    <xs:element name="BodyType" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="DeviceInformation" minOccurs="0">
    <xs:complexType>
      <xs:all>
        <xs:element name="Status" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="DevicePassword" minOccurs="0">
    <xs:complexType>
      <xs:all>
        <xs:element name="Status" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="UserInformation" minOccurs="0">
    <xs:complexType>
      <xs:all>
        <xs:element name="Status" type="xs:integer" minOccurs="0"/>
        <xs:element name="Get" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="Accounts" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Account" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:all>
```



Copyright © 2010 Microsoft Corporation.

```
                                <xs:element name="AccountId" type="xs:string" minOccurs="0"/>
                                <xs:element name="AccountName" type="xs:string"
minOccurs="0"/>
                                <xs:element name="UserDisplayName" type="xs:string"
minOccurs="0"/>
                                <xs:element name="SendDisabled" type="xs:boolean"
minOccurs="0"/>
                                <xs:element name="EmailAddresses" minOccurs="0">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="SMTPAddress" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
                                      <xs:element name="PrimarySmtpAddress" type="xs:string"
minOccurs="0"/>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:all>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:all>
                </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="RightsManagementInformation" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="Status" type="xs:integer" minOccurs="0"/>
              <xs:element name="Get">
                <xs:complexType>
                  <xs:all>
                    <xs:element ref="rm:RightsManagementTemplates"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### 2.2.2.16.2.1 Settings

The <Settings> element is the top-level element in the XML stream for the **Settings** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Oof><br><DeviceInformation><br><DevicePassword> | **Container** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <UserInformation><br><Status> (response only) | | |

The <Settings> element encapsulates one or more named property nodes that contain actions and arguments that apply to those properties.

#### 2.2.2.16.2.1.1 Status

The <Status> element contains a code that indicates the success or failure of the **Settings** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | None | **Integer** | 1 (required) |

The following table lists the valid values for the <Status> element in the context of the **Settings** command response. This is the status at the top level.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

Error code values 100 to 255 are reserved for property-specific error codes and vary from property to property. Any status value that is not 1 is a failure.

#### 2.2.2.16.2.1.2 Oof

The <Oof> element specifies a named property node for retrieving and setting OOF information.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Get><br><Set> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |

The **Settings** command supports <Get> and <Set> operations for the <Oof> element. The <Oof> element enables a user to do the following:

- Specify whether the user is currently out of office.
- Schedule an out of office message to be sent between a particular start date and particular end date.
- Specify the message that is to be shown to various audiences when the mobile device user is out of office.

OOF Get Request and Response



Copyright © 2010 Microsoft Corporation.

The <Get> element within the <Oof> element enables the client to retrieve OOF information from the server. The client specifies the <BodyType> to be retrieved and the server will return all OOF information and messages.

There is one <OofMessage> node per audience in an OOF <Get> response. If the sender group is allowed, but is disabled and has no Reply message (specified by the <ReplyMessage> element), an <OofMessage> node is still reported to the client.

If the client does not receive a group, it is presumably because the client does not have permission to enter settings for that group; in such a case, any attempt to set those properties MUST result in an Access Denied status code.

OOF Set Request and Response

The <Set> element enables the client to set the OOF status, time OOF, and OOF messages for one or more of the following groups:

- Internal
- External Known Senders (such as contacts)
- External Unknown Senders

#### 2.2.2.16.2.1.2.1 Status

The <Status> element contains a code that indicates the success or failure of the <Oof> <Get> or <Set> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Oof> | None | **Integer** | 0…1 (optional) |

The following table lists the values for <Status> in a **Settings** command <Oof> response.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. The XML code is formatted incorrectly. |

If <Oof> nodes <AppliesToExternalKnown> or <AppliesToExternalUnknown> are not allowed and are disabled by the administrator but are sent by the client in the <Set> request, <Oof> returns a successful <Status> value of 1 even though the user does not have access to these settings.

For <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.16.2.1.2.2 Get

The <Get> element contains <Oof> information retrieved from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Oof> | <BodyType> (request only)<br><OofState> (response only)<br><StartTime> (response only) | <Container> | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <EndTime> (response only)<br><OofMessage> (response only) | | |

#### 2.2.2.16.2.1.2.2.1 OofState

The <OofState> element specifies the availability of the <OOF> property.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | None | **Integer** | 0...1 (optional) |

The following table lists the valid values for <OofState>.

| Value | Meaning |
|---|---|
| 0 | The <Oof> property is disabled. |
| 1 | The <Oof> property is global. |
| 2 | The <Oof> property is time-based. |

<OofState> MUST be set to 2 if the <StartTime> and <EndTime> elements are present. If <OofState> is not set to 2 and the <StartTime> and <EndTime> elements are submitted in the request, the client does receive a successful response message, but the server does not store the <StartTime> and <EndTime> values.

#### 2.2.2.16.2.1.2.2.2 StartTime

The <StartTime> element is used with the <EndTime> element to specify the range of time during which the user is OOF.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | None | **DateTime** | 1...1 (required) |

The <StartTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

In a <Set> <Oof> node, both <StartTime> and <EndTime> MUST be included in the **Settings** request, or neither <StartTime> or <EndTime> MUST be included in the **Settings** request. If either <StartTime> or <EndTime> is included in the request without the other, a <Status> value of 2 is returned as a child of the <Oof> element.

#### 2.2.2.16.2.1.2.2.3 EndTime

The <EndTime> element is used with the <StartTime> element to specify the range of time during which the user is OOF.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | None | **DateTime** | 1...1 (required) |

The <EndTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

In a <Set> <Oof> node, both <StartTime> and <EndTime> MUST be included in the **Settings** request, or neither <StartTime> or <EndTime> MUST be included in the **Settings** request. If either <StartTime> or <EndTime> is included in the request without the other, a <Status> value of 2 is returned as a child of the <Oof> element.

#### 2.2.2.16.2.1.2.2.4 OofMessage

The <OofMessage> element contains a set of elements that specify the **OOF message** for a particular audience.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (<Oof> response only)<br><Set> (<Oof> request only) | <AppliesToInternal><br><AppliesToExternalKnown><br><AppliesToExternalUnknown><br><Enabled><br><ReplyMessage><br><BodyType> | **Container** | 0...3 |

The <OofMessage> element supports the following three audiences:<55>

- Internal—A user who is in the same organization as the sending user.
- Known external—A user who is outside the sending user's organization, but is represented in the sending user's contacts.
- Unknown external—A user who is outside the sending user's organization and is not represented in the sending user's contacts.

The presence of one of the following elements, which are mutually exclusive, indicates the audience to which an OOF message pertains:

- <AppliesToInternal>—The OOF message is relevant to an internal audience.
- <AppliesToExternalKnown>—The OOF message is relevant to a known external audience.
- <AppliesToExternalUnknown>—The OOF message is relevant to an unknown external audience.

There is one <OofMessage> node per audience in an OOF <Get> response. If a sender group is allowed, but is disabled and has no reply message (specified by the <ReplyMessage> element), an <OofMessage> node is reported to the client. If <AppliesToExternalKnown> or <AppliesToExternalUnknown> are not allowed and are disabled by the administrator but are sent by the client in the <Set> request, <Set> returns a successful <Status> value of 1 even though the user does not have access to these settings. Similarly, <AppliesToExternalKnown> and <AppliesToExternalUnknown> are returned to the client in a <Get> response even if the sender group is not allowed and is disabled.



Copyright © 2010 Microsoft Corporation.

In an <Oof> <Set> request, the client MUST NOT include the same AppliesTo* element in more than one <OofMessage> element.

#### 2.2.2.16.2.1.2.2.4.1 AppliesToInternal

The <AppliesToInternal> element indicates that the OOF message applies to internal users. (An internal user is a user who is in the same organization as the sending user.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |

The <AppliesToInternal> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToInternal/> tag.

When the <AppliesToInternal> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to internal users.

The following are the peer elements of the <AppliesToInternal> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF message itself.
- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.2.1.2.2.4.2 AppliesToExternalKnown

The <AppliesToExternalKnown> element indicates that the OOF message applies to known external users. (A known external user is a user who is outside the sending user's organization, but is represented in the sending user's contacts.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |

The <AppliesToExternalKnown> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalKnown/> tag.

When the <AppliesToExternalKnown> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to known external users.

The following are the peer elements of the <AppliesToExternalKnown> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.



Copyright © 2010 Microsoft Corporation.

- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.2.1.2.2.4.3 AppliesToExternalUnknown

The <AppliesToExternalUnknown> element indicates that the OOF message applies to unknown external users. (An unknown external user is a user who is outside the sending user's organization and is not represented in the sending user's contacts.)

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Empty tag** | 0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>) |

The <AppliesToExternalUnknown> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalUnknown/> tag.

When the <AppliesToExternalUnknown> element is present, its peer elements (that is, the other elements within the <OofMessage> element) specify the OOF settings with regard to unknown external users.

The following are the peer elements of the <AppliesToExternalUnknown> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

#### 2.2.2.16.2.1.2.2.4.4 Enabled

The <Enabled> element specifies whether an OOF message is sent to this audience while the sending user is OOF.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **Integer** | 0...1 (optional) |

The <Enabled> element is used in the OOF <Get> response to retrieve the current value. The <Enabled> element is used in the OOF <Set> request to set the value.

The value of <Enabled> is 1 if an OOF message is sent while the sending user is OOF; otherwise, the value is 0.

#### 2.2.2.16.2.1.2.2.4.5 ReplyMessage

The <ReplyMessage> element specifies the message to be shown to a particular audience when the user is OOF.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <OofMessage> | None | **String** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The <ReplyMessage> can be used in an OOF <Get> response to convey the requested OOF message, or in an OOF <Set> request to set the message that the client wants to send to a particular audience.

#### 2.2.2.16.2.1.2.2.4.6 BodyType

The <BodyType> element is a string that specifies the format of the OOF message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (request only)<br><OofMessage> | None | **String** | 1...1 (required) |

The following are the permitted values for the <BodyType> element:

- *Text*
- *HTML*

If <BodyType> has the value of **HTML**, all message strings are sent in the HTML format. If <BodyType> has the value *Text*, the message strings are sent in plain text. Because there is no default value, the <BodyType> node MUST be present.

### 2.2.2.16.2.1.3 DeviceInformation

The <DeviceInformation> element is the container node that is used for sending the client device's properties of the client device to the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Set> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |

Clients SHOULD use the **Provision** command to send <DeviceInformation> to the server.<56>

When the **Settings** command is used to send the <DeviceInformation> element, it sends the following information about a client device to the server:

- Device model
- International Mobile Equipment Identity (IMEI)
- Device friendly name
- Device operating system
- Telephone number
- Device operating system language
- User Agent
- Whether to enable outbound SMS (see [MS-ASMS])
- Mobile operator name



Copyright © 2010 Microsoft Corporation.

The device information is represented as a flat list of settings under the <DeviceInformation> node in the **Settings** command. <DeviceInformation> has only one child element, <Set>, which contains the list of device information items in the request. The <DeviceInformation> property supports only the <Set> operation because this information is write-only from the device.

#### 2.2.2.16.2.1.3.1 Status

The <Status> element contains a code that indicates the success or failure of the <DeviceInformation> <Set> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <DeviceInformation> | None | **Integer** | 1 (required) |

The following table lists the valid values for <Status> in a **Settings** command <DeviceInformation> response.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. The XML code is formatted incorrectly. |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.16.2.1.4 DevicePassword

The <DevicePassword> element is a container node that is used to send the recovery password of the client device to the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Set> (request only)<br><Status> (response only) | **Container** | 0...1 (optional) |

Use the <Set> operation on the <DevicePassword> property to enable the device to send or store a recovery password on the server. The recovery password is stored in the user's mailbox and can be retrieved by the administrator or the end-user if the user forgets his or her password.

#### 2.2.2.16.2.1.4.1 Status

The <Status> element contains a code that indicates the success or failure of the <DevicePassword> <Set> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <DevicePassword> | None | **Integer** | 1 (required) |

The following table lists the values for <Status> in a **Settings** command <DevicePassword> response.



Copyright © 2010 Microsoft Corporation.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. The XML code is formatted incorrectly. |
| 5 | Invalid arguments. The specified password is too long. |
| 7 | Denied by policy. The administrator has disabled password recovery in this deployment. |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.16.2.1.5 UserInformation

The <UserInformation> element is a container node that is used to request a list of a user's e-mail addresses from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> | <Get><br><Status> (<UserInformation> response only) | **Container** | 0...1 (optional) |

The list of a user's e-mail addresses can be useful, for example, for ensuring that the user is not included when performing a Reply to All operation to an e-mail message.

In a request, the <UserInformation> element contains the <Get> element to indicate that the server is to return all available e-mail addresses for the user.

The **Settings** command supports read-only access to the list of a user's various e-mail addresses via the <Get> element. The client is unable to write this information.

##### 2.2.2.16.2.1.5.1 Status

The <Status> element contains a code that indicates the success or failure of the <UserInformation> <Get> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <UserInformation> | None | **Integer** | 1 (required) |

The following table lists the values for <Status> in a **Settings** command <UserInformation> response.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. The XML code is formatted incorrectly. |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

##### 2.2.2.16.2.1.5.2 Get

The <Get> element contains <Oof> information retrieved from the server.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <UserInformation> | None (request only)<br><EmailAddresses> (response only)<br><Accounts> (response only) | **Empty tag** (request only)<br>**Container** (response only) | 0…1 (optional) |

#### 2.2.2.16.2.1.5.2.1 Accounts

The <Accounts> element<57> is a container for all aggregate accounts that the user subscribes to.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> | <Account> | **Container** | 0…1 (optional) |

##### 2.2.2.16.2.1.5.2.1.1 Account

The <Account> element<58> is a container for all account information associated with a single account.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Accounts> | <AccountId><br><AccountName><br><UserName><br><EmailAddresses> | **Container** | 1…N (required) |

###### 2.2.2.16.2.1.5.2.1.1.1 AccountId

The <AccountId> element<59> specifies the ID of the given account.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Account> | None | **String** (up to 64 characters) | 0…1 (optional) |

The primary account, as identified by the <PrimarySmtpAccount> element, does not have an <AccountId> value.

###### 2.2.2.16.2.1.5.2.1.1.2 AccountName

The <AccountName> element<60> specifies the friendly name for the given account.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Account> | None | **String** (up to 512 characters) | 0…1 (optional) |

###### 2.2.2.16.2.1.5.2.1.1.3 UserDisplayName

The <UserDisplayName> element<61> specifies the display name of the user associated with the given account.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Account> | None | **String** (up to 512 characters) | 0…1 (optional) |

#### 2.2.2.16.2.1.5.2.1.1.4 SendDisabled

The <SendDisabled> element<62> specifies whether the client can send messages using the given account.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Account> | None | **Boolean** | 0…1 (optional) |

True if the client cannot send using the given account; otherwise, false.

#### 2.2.2.16.2.1.5.2.1.1.5 EmailAddresses

The <EmailAddresses> element contains one or more e-mail addresses for the user.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Get> (response only) | <SMTPAddress> (response only) | **Container** | 0...1 (optional) |

#### 2.2.2.16.2.1.5.2.1.1.5.1 SMTPAddress

The <SmtpAddress> element specifies one of the user's e-mail addresses.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmailAddresses> (response only) | None | **String** | 1...N (optional) |

#### 2.2.2.16.2.1.5.2.1.1.5.2 PrimarySmtpAddress

The <PrimarySmtpAddress> element<63> specifies the primary SMTP address for the given account.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <EmailAddresses> | None | **String** | 0…1 (optional) |

The value of the <PrimarySmtpAddress> element can also be returned as a value for the <SmtpAddress> element.

### 2.2.2.16.2.1.6 RightsManagementInformation

The <RightsManagementInformation> element<64> is a container node that contains rights management information settings retrieved from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Settings> (request and response) | <Get> (request or response)<br><Status> (response only) | **Container** | 0…1 (optional) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.16.2.1.6.1 Status

The <Status> element contains a code that indicates the success or failure of the <RightsManagementInformation> <Get> operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <RightsManagementInformation> | None | **Integer** | 1 (required) |

The following table lists the valid values for <Status> in a **Settings** command <RightsManagementInformation> response.

| Value | Meaning |
|---|---|
| 1 | Success. |
| 2 | Protocol error. The XML code is formatted incorrectly. |

For <Status> values common to all ActiveSync commands, see section 2.2.3.

#### 2.2.2.16.2.1.6.2 Get

The <Get> element contains <RightsManagementInformation> settings retrieved from the server.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <RightsManagementInformation> | None (request only)<br><rm:RightsManagementTemplates> as specified in [MS-ASRM] section 2.2.2.4 (response only) | **Empty tag** (request only)<br>**Container** (response only) | 1 (required) |

### 2.2.2.17 SmartForward

The **SmartForward** command is used by clients to forward messages without retrieving the full, original message from the server.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, messages can use the <SaveInSentItems> element to automatically have the messages saved on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command. Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

The **SmartForward** command can be applied to a meeting. When **SmartForward** is applied to a recurring meeting, the <InstanceId> element specifies the ID of a particular occurrence in the recurring meeting. If **SmartForward** is applied to a recurring meeting and the <InstanceId> element is absent, the server SHOULD forward the entire recurring meeting. If the value of the <InstanceId> element is invalid, the server responds with <Status> value 104, as specified in section 2.2.3.



Copyright © 2010 Microsoft Corporation.

When **SmartForward** is applied to an appointment, the original message is included by the server as an attachment to the outgoing message. When smart-forwarding a normal message or a meeting, **SmartForward's** behavior is the same as that of the **SmartReply** command.

The **SmartForward** command is similar to the **SendMail** command, but the outgoing message identifies the item being forwarded to and includes the text of the new message. The full text of the original message is retrieved and sent by the server. Using the server copy of the original message saves network bandwidth by not downloading the original message to the client and then uploading it again with the forward.

The **SmartForward** command lists the message recipients.

By default, because the original message and the forward messages can use different **character sets**, this command will always send the outgoing message by using the UTF8 character set for the body of the message.

The ComposeMail namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ComposeMail namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.17.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **SmartForward** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:rm="RightsManagement:">

    <xs:import namespace="RightsManagement:"/>

    <xs:complexType name="EmptyTag" />

    <xs:element name="SmartForward">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
                            <xs:maxLength value="40"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Source">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="FolderId" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="1"/>
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
                        </xs:element>
                        <xs:element name="ItemId" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:minLength value="1"/>
                                    <xs:maxLength value="64"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="LongId" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:minLength value="1"/>
                                    <xs:maxLength value="256"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="InstanceId" type="xs:dateTime" minOccurs="0"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="AccountId" type="xs:string" minOccurs="0"/>
            <xs:element name="SaveInSentItems" type="tns:EmptyTag" minOccurs="0"/>
            <xs:element name="ReplaceMime" type="tns:EmptyTag" minOccurs="0"/>
            <xs:element name="Mime" type="xs:string"/>
            <xs:element ref="rm:TemplateID" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:schema>
```

#### 2.2.2.17.1.1 SmartForward

The <SmartForward> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartForward** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><Source> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><ReplaceMime> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS-ASRM] section 2.2.2.2<br><Status> (response only) | **Container** | 1…1 (required) |

##### 2.2.2.17.1.1.1 ClientId

The <ClientId> element specifies the client's unique message ID (MID).



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (request only) | None | **String** (Up to 40 characters) | 1...1 (required) |

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.

#### 2.2.2.17.1.1.2 Source

The <Source> element contains information about the source message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (request only) | <FolderId> (request only)<br><ItemId> (request only)<br><LongId> (request only)<br><InstanceId> (request only) | **Container** | 1...1 (required) |

##### 2.2.2.17.1.1.2.1 FolderId

The <FolderId> element contains the **folder ID (FID)** for the source message, which is returned in the **FolderSync** command response message. If the <FolderId> element is present, the <ItemId> element MUST also be present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 64 characters) | 0...1 (optional) |

##### 2.2.2.17.1.1.2.2 ItemId

The <ItemId> element contains the item ID for the source message, which is returned in the **Sync** command response message. If the <ItemId> element is present, the <FolderId> element MUST also be present if the message being forwarded is stored in a folder other than the Inbox folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 64 characters) | 0...1 (optional) |

##### 2.2.2.17.1.1.2.3 LongId

The <LongId> element contains the long ID for the source message, which is returned in the **Search** command response message. If the <LongId> element is present, neither the <FolderId> or<ItemId> elements are present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 256 characters) | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.17.1.1.2.4 InstanceId

The <InstanceId> element contains the instance of a recurrence for the source item. If the <InstanceId> element is present, both the <FolderId> and <ItemId> elements SHOULD be present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **DateTime** | 0...1 (optional) |

The format of the <InstanceId> element is a **dateTime** value that includes the punctuation separators. For example, 2010-03-20T22:40:00.000Z.

#### 2.2.2.17.1.1.3 AccountId

The <AccountId> element<65> identifies the account from which an e-mail is sent.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> | None | **String** | 0...1 (optional) |

If the <AccountId> element is not present in the **SmartForward** request, the server sends the e-mail using the account identified by the <PrimarySmtpAddress> element returned in the **Settings** command response.

If <AccountId> is included in the request, the value MUST equal one of the <AccountId> values included in the **Settings** command response (section 2.2.2.16.2).

The server MUST validate that the <AccountId> value provided is valid for sending e-mail. A <Status> value of 166 is returned if the <AccountId> value is not valid. A <Status> value of 167 is returned if the account does not support sending e-mail.

**Note** The server sends the e-mail by using the <AccountId> value specified by the <AccountId> element, and not the account specified by the <From> element.

#### 2.2.2.17.1.1.4 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should be stored in the Sent Items folder. If the <SaveInSentItems> element is present, the message is stored -- if not present, the message is not stored.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (request only) | None | **Empty tag** | 0...1 (optional) |

The <SaveInSentItems> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <SaveInSentItems/> tag.

#### 2.2.2.17.1.1.5 ReplaceMime

The <ReplaceMime> element specifies whether the client is sending the entire message or not. When <ReplaceMime> is present, the server MUST not include the body or attachments of the original message being forwarded. When not included, the client MUST append the body of the original message as attachments to the outgoing message.



Copyright © 2010 Microsoft Corporation.

The client can use this tag to indicate whether the message was edited inline, or whether the message had reply/forward text prepended to the source message. If the <ReplaceMime> element is present, the message was edited.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (request only) | None | **Empty tag** | 0...1 (optional) |

The <ReplaceMime> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <ReplaceMime/> tag.

#### 2.2.2.17.1.1.6 Mime

The <Mime> element contains the MIME-encoded message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (request only) | None | **Byte array** ([MS-ASDTYPE] section 2.6.1) | 1...1 (required) |

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

### 2.2.2.17.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **SmartForward** command response.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="SmartForward:" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="SmartForward:" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="SmartForward">
      <xs:complexType>
         <xs:all>
            <xs:element name="Status" type="xs:integer" minOccurs="0"/>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

If the message was forwarded successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```

#### 2.2.2.17.2.1 SmartForward

The <SmartForward> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartForward** command.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><Source> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><ReplaceMime> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS-ASRM] section 2.2.2.2<br><Status> (response only) | **Container** | 1...1 (required) |

#### 2.2.2.17.2.1.1 Status

The <Status> element indicates the reason for the failure of a **SmartForward** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartForward> (response only) | None | **Integer** | 0...1 (optional) |

If the command succeeds, no XML body is returned in the response, as specified in section 2.2.2.17.2. If the command fails, the <Status> element contains a code that indicates the type of failure.

Valid <Status> values are listed in section 2.2.3.

### 2.2.2.18 SmartReply

The **SmartReply** command is used by clients to reply to messages without retrieving the full, original message from the server.

The **SmartReply** command is similar to the **SendMail** command, but the outgoing message identifies the item being replied to and includes the text of the new message. The full text of the original message is retrieved and sent by the server. Using the server copy of the original message saves network bandwidth by not downloading the original message to the client and then uploading it again with the reply.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, messages can use the <SaveInSentItems> element to automatically have the messages saved on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command. Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

The **SmartReply** command can be applied to a meeting. When **SmartReply** is applied to a recurring meeting, the <InstanceId> element specifies the ID of a particular occurrence in the recurring meeting. If **SmartReply** is applied to a recurring meeting and the <InstanceId> element is absent, the server SHOULD reply for the entire recurring meeting. If the value of the <InstanceId> element is invalid, the server responds with Status value 104, as specified in section 2.2.3.



Copyright © 2010 Microsoft Corporation.

The **SmartReply** command lists the message recipients, so it is used to implement both Reply and Reply-to-All functionality. It is the responsibility of the client to implement Reply and Reply-to-All functionality.

By default, because the original message and the reply messages can use different character sets, this command will always send the outgoing message by using the UTF8 character set for the body of the message.

The ComposeMail namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ComposeMail namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.18.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **SmartReply** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:rm="RightsManagement:">

    <xs:import namespace="RightsManagement:"/>

    <xs:complexType name="EmptyTag" />

    <xs:element name="SmartReply">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
                            <xs:maxLength value="40"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Source">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="FolderId" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="1"/>
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="ItemId" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:minLength value="1"/>
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
```



Copyright © 2010 Microsoft Corporation.

```
                        </xs:element>
                        <xs:element name="LongId" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:minLength value="1"/>
                                    <xs:maxLength value="256"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="InstanceId" type="xs:string" minOccurs="0"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="AccountId" type="xs:string" minOccurs="0"/>
            <xs:element name="SaveInSentItems" type="tns:EmptyTag" minOccurs="0"/>
            <xs:element name="ReplaceMime" type="tns:EmptyTag" minOccurs="0"/>
            <xs:element name="Mime" type="xs:string"/>
            <xs:element ref="rm:TemplateID" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:schema>
```

#### 2.2.2.18.1.1 SmartReply

The <SmartReply> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartReply** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><Source> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><ReplaceMime> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS-ASRM] section 2.2.2.2<br><Status> (response only) | **Container** | 1...1 (required) |

##### 2.2.2.18.1.1.1 ClientId

The required element <ClientId> specifies the client's unique message ID (MID).

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (request only) | None | **String** (Up to 40 characters) | 1...1 (required) |

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.18.1.1.2 Source

The <Source> element contains information about the source message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (request only) | <FolderId> (request only)<br><ItemId> (request only)<br><LongId> (request only)<br><InstanceId> (request only) | **Container** | 1...1 (required) |

##### 2.2.2.18.1.1.2.1 FolderId

The <FolderId> element contains the folder ID (FID) for the source message, which is returned in the **FolderSync** command response message. If the <FolderId> element is present, the <ItemId> element MUST also be present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 64 characters) | 0...1 (optional) |

##### 2.2.2.18.1.1.2.2 ItemId

The <ItemId> element contains the item ID for the source message, which is returned in the **Sync** command response message. If the <ItemId> element is present, the <FolderId> element MUST also be present if the message being replied to is stored in a folder other than the Inbox folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 64 characters) | 0...1 (optional) |

##### 2.2.2.18.1.1.2.3 LongId

The <LongId> element contains the long ID for the source message, which is returned in the **Search** command response message. If the <LongId> element is present, none of the <FolderId>, <ItemId>, or <InstanceId> elements is present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> (request only) | None | **String** (up to 256 characters) | 0...1 (optional) |

##### 2.2.2.18.1.1.2.4 InstanceId

The <InstanceId> element contains the instance of a recurrence for the source item. If the <InstanceId> element is present, both the <FolderId> and <ItemId> elements SHOULD be present.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Source> | None | **DateTime** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The format of the <InstanceId> element is a **dateTime** value that includes the punctuation separators. For example, 2010-03-20T22:40:00.000Z.

#### 2.2.2.18.1.1.3 AccountId

The <AccountId> element<66> identifies the account from which an e-mail is sent.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SendMail> | None | **String** | 0...1 (optional) |

If the <AccountId> is not present in the **SmartReply** request, the server sends the e-mail using the account identified by the <PrimarySmtpAddress> element returned in the **Settings** command response.

If <AccountId> is included in the request, the value MUST equal one of the <AccountId> values included in the **Settings** command response (section 2.2.2.16.2).

The server MUST validate that the <AccountID> provided is valid for sending email. A <Status> value of 166 is returned if the <AccountId> is not valid. A <Status> value of 167 is returned if the account does not support sending e-mail.

**Note** The server sends the e-mail using the <AccountId> specified by the <AccountId> element, and not the account specified by the <From> element.

#### 2.2.2.18.1.1.4 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should be stored in the Sent Items folder. If the <SaveInSentItems> element is present, the message is stored – if not present, the message is not stored.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (request only) | None | **Empty tag** | 0...1 (optional) |

The <SaveInSentItems> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <SaveInSentItems/> tag.

#### 2.2.2.18.1.1.5 ReplaceMime

The <ReplaceMime> element specifies whether the client is sending the entire message or not. When <ReplaceMime> is present, the server MUST not include the body or attachments of the original message being forwarded. When not included, the client MUST append the body of the original message as attachments to the outgoing message.

The client can use this tag to indicate whether the message was edited inline, or whether the message had reply/forward text prepended to the source message. If the <ReplaceMime> element is present, the message was edited inline.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (request only) | None | **Empty tag** | 0...1 (optional) |



Copyright © 2010 Microsoft Corporation.

The <ReplaceMime> element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <ReplaceMime/> tag.

#### 2.2.2.18.1.1.6 Mime

The <Mime> element contains the MIME-encoded message.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (request only) | None | **Byte array** ([MS-ASDTYPE] section 2.6.1) | 1...1 (required) |

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

### 2.2.2.18.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **SmartReply** command response.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="SmartReply:" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="SmartReply:" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="SmartReply">
      <xs:complexType>
         <xs:all>
            <xs:element name="Status" type="xs:integer" minOccurs="0"/>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

If the message was sent successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```

#### 2.2.2.18.2.1 SmartReply

The <SmartReply> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartReply** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <ClientId> (request only)<br><Source> (request only)<br><AccountId> (request only)<br><SaveInSentItems> (request only)<br><ReplaceMime> (request only)<br><Mime> (request only)<br><rm:TemplateID> (request only) as specified in [MS- | **Container** | 1...1 (required) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | ASRM] section 2.2.2.2<br><Status> (response only) | | |

#### 2.2.2.18.2.1.1 Status

The <Status> element indicates the reason for the failure of a **SmartReply** command request.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SmartReply> (response only) | None | **Integer** | 0...1 (optional) |

If the command succeeds, no XML body is returned in the response, as specified in section 2.2.2.18.2. If the command fails, the <Status> element contains a code that indicates the type of failure.

Valid <Status> values are listed in section 2.2.3. In particular, a <Status> value of 117 indicates that the server does not allow a reply to the message.

### 2.2.2.19 Sync

The **Sync** command synchronizes changes in a collection between the client and the server.

The AirSync namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the AirSync namespace use the namespace prefixes defined in section 2.2.1.

For more details about the AirSyncBase elements that are used by this command, see [MS-ASAIRS] section 2.2.

Synchronization requires a priming of the system; therefore for each collection that the client wishes to synchronize, it MUST issue an initial **Sync** request by sending a synchronization key of 0. This request establishes a synchronization relationship with the server and initializes the synchronization state there. The server responds with an initial value of the synchronization key, which the client MUST then use to get the initial set of objects from the server. (From this point forward, client requests MUST always include the synchronization key that was received in the last response from the server.) The client then sends a **Sync** command request to the server with the response synchronization key and includes any changes that were made on the client.

If the client device has not yet synchronized a folder, there SHOULD be no client-side changes. The device MUST synchronize the full contents of a given folder, and then have its changes, additions, and deletions applied.

The response from the server indicates whether the client's changes were accepted, and includes any changes that were made on the server. The server response also contains a synchronization key that is to be used for the next synchronization session for the folder.

This protocol has been optimized for the case in which there are no changes to any of the collections that are specified in the **Sync** request. In such a case, the client can receive an empty response from the server. After the client receives an empty response, the client can issue an empty **Sync** request. The server then re-executes the previous request, which it cached.



Copyright © 2010 Microsoft Corporation.

Certain ActiveSync classes support **ghosted** properties. A ghosted property whose value has not changed from the last **Sync** response can be excluded from the request body, and its value on the server will be preserved instead of being deleted. A client uses the <Supported> element to specify to the server which properties are managed by the client and not ghosted by the server. For more information, see section 2.2.2.19.1.2.1.1.3.

The following diagram shows request and response processing by the client.

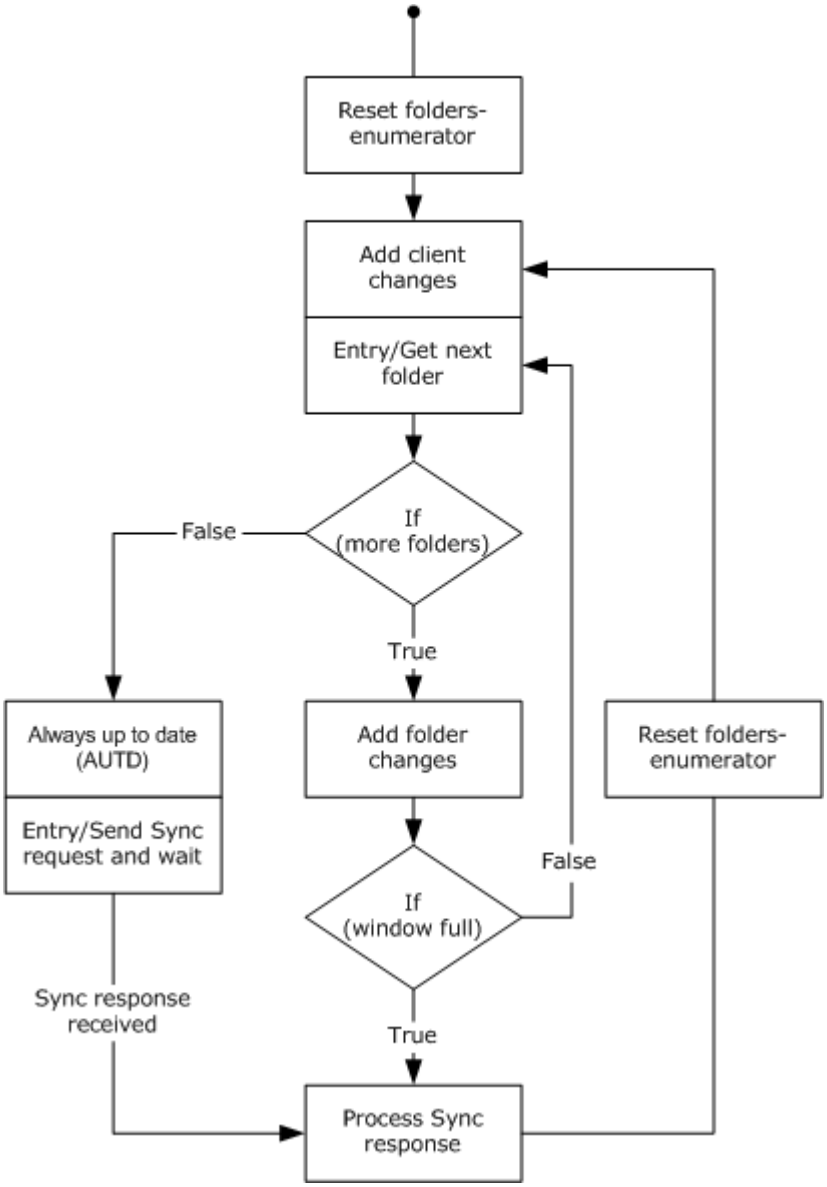


**Figure 3: Sync command client processing**

### 2.2.2.19.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **Sync** command request.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="AirSync:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="AirSync:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:contacts="Contacts:"
        xmlns:contacts2="Contacts2:"
        xmlns:calendar="Calendar:"
        xmlns:email="Email:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:tasks="Tasks:"
        xmlns:notes="Notes:"
        xmlns:rm="RightsManagement:">


        <xs:import namespace="Contacts2:"/>
        <xs:import namespace="Contacts:"/>
        <xs:import namespace="Email:"/>
        <xs:import namespace="Calendar:"/>
        <xs:import namespace="AirSyncBase:"/>
        <xs:import namespace="Tasks:"/>
        <xs:import namespace="Notes:"/>
        <xs:import namespace="RightsManagement:"/>


<xs:complexType name="EmptyTag" />

  <xs:element name="Sync">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Collections" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="Collection">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="SyncKey">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="CollectionId">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element minOccurs="0" name="Supported">
                      <xs:complexType mixed="true">
                        <xs:sequence minOccurs="0">
                          <xs:choice maxOccurs="unbounded">
                            <xs:group ref="contacts:GhostingProps"/>
                            <xs:group ref="contacts2:GhostingProps"/>
                            <xs:group ref="calendar:GhostingProps"/>
```



Copyright © 2010 Microsoft Corporation.

```
                        </xs:choice>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element minOccurs="0" name="DeletesAsMoves" type="xs:boolean"/>
                  <xs:element minOccurs="0" name="GetChanges" type="xs:boolean"/>
                  <xs:element minOccurs="0" name="WindowSize">
                    <xs:simpleType>
                      <xs:restriction base="xs:integer">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="512"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="ConversationMode" minOccurs="0" type="xs:boolean"/>
                  <xs:element minOccurs="0" maxOccurs="2" name="Options">
                    <xs:complexType>
                      <xs:choice maxOccurs="unbounded">
                        <xs:element name="FilterType" minOccurs="0">
                          <xs:simpleType>
                            <xs:restriction base="xs:unsignedByte">
                              <xs:minInclusive value="0"/>
                              <xs:maxInclusive value="8"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element name="Class" type="xs:string" minOccurs="0"/>
                        <xs:element ref="airsyncbase:BodyPreference" minOccurs="0"
maxOccurs="unbounded" />
                        <xs:element ref="airsyncbase:BodyPartPreference" minOccurs="0"/>
                        <xs:element minOccurs="0" name="Conflict" type="xs:unsignedByte">
                            <xs:simpleType>
                                <xs:restriction base="xs:unsignedByte">
|                                    <xs:minInclusive value="0"/>
                                   <xs:maxInclusive value="1"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element minOccurs="0" name="MIMESupport">
                          <xs:simpleType>
                            <xs:restriction base="xs:unsignedByte">
                              <xs:minInclusive value="0" />
                              <xs:maxInclusive value="2" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element minOccurs="0" name="MIMETruncation">
                          <xs:simpleType>
                            <xs:restriction base="xs:unsignedByte">
                              <xs:minInclusive value="0" />
                              <xs:maxInclusive value="8" />
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element name="MaxItems" minOccurs="0">
                          <xs:simpleType>
                            <xs:restriction base="xs:integer">
                              <xs:minInclusive value="1"/>
                            </xs:restriction>
```



Copyright © 2010 Microsoft Corporation.

```
                    </xs:simpleType>
                  </xs:element>
                  <xs:element ref="rm:RightsManagementSupport" minOccurs="0" />
                </xs:choice>
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" name="Commands">
              <xs:complexType>
                <xs:choice maxOccurs="unbounded">
                  <xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="ServerId">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:maxLength value="64"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element name="ApplicationData">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:choice maxOccurs="unbounded">
                                <xs:element ref="email:Flag"/>
                                <xs:element ref="email:Read"/>
                                <xs:element ref="email:Categories"/>
                                <xs:element ref="calendar:OrganizerName"/>
                                <xs:element ref="calendar:OrganizerEmail"/>
                                <xs:element ref="calendar:Exceptions"/>
                                <xs:element ref="calendar:Attendees"/>
                                <xs:element ref="calendar:DisallowNewTimeProposal"/>
                                <xs:element ref="calendar:ResponseRequested"/>
                                <xs:element ref="calendar:TimeZone"/>
                                <xs:element ref="calendar:AllDayEvent"/>
                                <xs:element ref="airsyncbase:NativeBodyType"/>
                                <xs:element ref="airsyncbase:Body"/>
                                <xs:element ref="calendar:BusyStatus"/>
                                <xs:element ref="calendar:Categories"/>
                                <xs:element ref="calendar:DtStamp"/>
                                <xs:element ref="calendar:EndTime"/>
                                <xs:element ref="calendar:Location"/>
                                <xs:element ref="calendar:MeetingStatus"/>
                                <xs:element ref="calendar:Reminder"/>
                                <xs:element ref="calendar:Sensitivity"/>
                                <xs:element ref="calendar:Subject"/>
                                <xs:element ref="calendar:StartTime"/>
                                <xs:element ref="calendar:UID"/>
                                <xs:element ref="calendar:Recurrence"/>
                                <xs:element ref="contacts:Anniversary"/>
                                <xs:element ref="contacts:AssistantName"/>
                                <xs:element ref="contacts:AssistantPhoneNumber"/>
                                <xs:element ref="contacts:AssistnamePhoneNumber"/>
                                <xs:element ref="contacts:Birthday"/>
                                <xs:element ref="contacts:Business2PhoneNumber"/>
                                <xs:element ref="contacts:BusinessAddressCity"/>
                                <xs:element ref="contacts:BusinessAddressCountry"/>
                                <xs:element
ref="contacts:BusinessAddressPostalCode"/>
                                <xs:element ref="contacts:BusinessAddressState"/>
```



Copyright © 2010 Microsoft Corporation.

```
<xs:element ref="contacts:BusinessAddressStreet"/>
<xs:element ref="contacts:BusinessFaxNumber"/>
<xs:element ref="contacts:BusinessPhoneNumber"/>
<xs:element ref="contacts:CarPhoneNumber"/>
<xs:element ref="contacts:Categories"/>
<xs:element ref="contacts:Children"/>
<xs:element ref="contacts:CompanyName"/>
<xs:element ref="contacts:Department"/>
<xs:element ref="contacts:Email1Address"/>
<xs:element ref="contacts:Email2Address"/>
<xs:element ref="contacts:Email3Address"/>
<xs:element ref="contacts:FileAs"/>
<xs:element ref="contacts:FirstName"/>
<xs:element ref="contacts:MiddleName"/>
<xs:element ref="contacts:Home2PhoneNumber"/>
<xs:element ref="contacts:HomeAddressCity"/>
<xs:element ref="contacts:HomeAddressCountry"/>
<xs:element ref="contacts:HomeAddressPostalCode"/>
<xs:element ref="contacts:HomeAddressState"/>
<xs:element ref="contacts:HomeAddressStreet"/>
<xs:element ref="contacts:HomeFaxNumber"/>
<xs:element ref="contacts:HomePhoneNumber"/>
<xs:element ref="contacts:JobTitle"/>
<xs:element ref="contacts:LastName"/>
<xs:element ref="contacts:MobilePhoneNumber"/>
<xs:element ref="contacts:OfficeLocation"/>
<xs:element ref="contacts:OtherAddressCity"/>
<xs:element ref="contacts:OtherAddressCountry"/>
<xs:element ref="contacts:OtherAddressPostalCode"/>
<xs:element ref="contacts:OtherAddressState"/>
<xs:element ref="contacts:OtherAddressStreet"/>
<xs:element ref="contacts:PagerNumber"/>
<xs:element ref="contacts:RadioPhoneNumber"/>
<xs:element ref="contacts:Spouse"/>
<xs:element ref="contacts:Suffix"/>
<xs:element ref="contacts:Title"/>
<xs:element ref="contacts:WebPage"/>
<xs:element ref="contacts:YomiCompanyName"/>
<xs:element ref="contacts:YomiFirstName"/>
<xs:element ref="contacts:Picture"/>
<xs:element ref="contacts2:CustomerId"/>
<xs:element ref="contacts2:GovernmentId"/>
<xs:element ref="contacts2:IMAddress"/>
<xs:element ref="contacts2:IMAddress2"/>
<xs:element ref="contacts2:IMAddress3"/>
<xs:element ref="contacts2:ManagerName"/>
<xs:element ref="contacts2:CompanyMainPhone"/>
<xs:element ref="contacts2:AccountName"/>
<xs:element ref="contacts2:NickName"/>
<xs:element ref="contacts2:MMS"/>
<xs:element ref="contacts:YomiLastName"/>
<xs:element ref="tasks:Complete"/>
<xs:element ref="tasks:Subject"/>
<xs:element ref="tasks:Categories"/>
<xs:element ref="tasks:DateCompleted"/>
<xs:element ref="tasks:DueDate"/>
<xs:element ref="tasks:UtcDueDate"/>
<xs:element ref="tasks:Importance"/>
<xs:element ref="tasks:Recurrence"/>
```



Copyright © 2010 Microsoft Corporation.

```
                <xs:element ref="tasks:ReminderSet"/>
                <xs:element ref="tasks:ReminderTime"/>
                <xs:element ref="tasks:Sensitivity"/>
                <xs:element ref="tasks:StartDate"/>
                <xs:element ref="tasks:UtcStartDate"/>
                <xs:element ref="notes:Subject"/>
                <xs:element ref="notes:MessageClass"/>
                <xs:element ref="notes:LastModifiedDate"/>
                <xs:element ref="notes:Categories"/>
              </xs:choice>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Delete">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ServerId">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="64"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Class" type="xs:string" minOccurs="0" />
        <xs:element name="ClientId">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="64"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="ApplicationData">
          <xs:complexType>
            <xs:sequence>
              <xs:choice maxOccurs="unbounded">
                <xs:element ref="calendar:OrganizerName"/>
                <xs:element ref="calendar:OrganizerEmail"/>
                <xs:element ref="calendar:Exceptions"/>
                <xs:element ref="calendar:Attendees"/>
                <xs:element ref="calendar:DisallowNewTimeProposal"/>
                <xs:element ref="calendar:ResponseRequested"/>
                <xs:element ref="calendar:TimeZone"/>
                <xs:element ref="calendar:AllDayEvent"/>
                <xs:element ref="airsyncbase:NativeBodyType"/>
                <xs:element ref="airsyncbase:Body"/>
                <xs:element ref="calendar:BusyStatus"/>
                <xs:element ref="calendar:Categories"/>
                <xs:element ref="calendar:DtStamp"/>
                <xs:element ref="calendar:EndTime"/>
                <xs:element ref="calendar:Location"/>
```



Copyright © 2010 Microsoft Corporation.

```
                                    <xs:element ref="calendar:MeetingStatus"/>
                                    <xs:element ref="calendar:Reminder"/>
                                    <xs:element ref="calendar:Sensitivity"/>
                                    <xs:element ref="calendar:Subject"/>
                                    <xs:element ref="calendar:StartTime"/>
                                    <xs:element ref="calendar:UID"/>
                                    <xs:element ref="calendar:Recurrence"/>
                                    <xs:element ref="contacts:Anniversary"/>
                                    <xs:element ref="contacts:AssistantName"/>
                                    <xs:element ref="contacts:AssistantPhoneNumber"/>
                                    <xs:element ref="contacts:AssistnamePhoneNumber"/>
                                    <xs:element ref="contacts:Birthday"/>
                                    <xs:element ref="contacts:Business2PhoneNumber"/>
                                    <xs:element ref="contacts:BusinessAddressCity"/>
                                    <xs:element ref="contacts:BusinessAddressCountry"/>
                                    <xs:element
  ref="contacts:BusinessAddressPostalCode"/>
                                    <xs:element ref="contacts:BusinessAddressState"/>
                                    <xs:element ref="contacts:BusinessAddressStreet"/>
                                    <xs:element ref="contacts:BusinessFaxNumber"/>
                                    <xs:element ref="contacts:BusinessPhoneNumber"/>
                                    <xs:element ref="contacts:CarPhoneNumber"/>
                                    <xs:element ref="contacts:Categories"/>
                                    <xs:element ref="contacts:Children"/>
                                    <xs:element ref="contacts:CompanyName"/>
                                    <xs:element ref="contacts:Department"/>
                                    <xs:element ref="contacts:Email1Address"/>
                                    <xs:element ref="contacts:Email2Address"/>
                                    <xs:element ref="contacts:Email3Address"/>
                                    <xs:element ref="contacts:FileAs"/>
                                    <xs:element ref="contacts:FirstName"/>
                                    <xs:element ref="contacts:MiddleName"/>
                                    <xs:element ref="contacts:Home2PhoneNumber"/>
                                    <xs:element ref="contacts:HomeAddressCity"/>
                                    <xs:element ref="contacts:HomeAddressCountry"/>
                                    <xs:element ref="contacts:HomeAddressPostalCode"/>
                                    <xs:element ref="contacts:HomeAddressState"/>
                                    <xs:element ref="contacts:HomeAddressStreet"/>
                                    <xs:element ref="contacts:HomeFaxNumber"/>
                                    <xs:element ref="contacts:HomePhoneNumber"/>
                                    <xs:element ref="contacts:JobTitle"/>
                                    <xs:element ref="contacts:LastName"/>
                                    <xs:element ref="contacts:MobilePhoneNumber"/>
                                    <xs:element ref="contacts:OfficeLocation"/>
                                    <xs:element ref="contacts:OtherAddressCity"/>
                                    <xs:element ref="contacts:OtherAddressCountry"/>
                                    <xs:element ref="contacts:OtherAddressPostalCode"/>
                                    <xs:element ref="contacts:OtherAddressState"/>
                                    <xs:element ref="contacts:OtherAddressStreet"/>
                                    <xs:element ref="contacts:PagerNumber"/>
                                    <xs:element ref="contacts:RadioPhoneNumber"/>
                                    <xs:element ref="contacts:Spouse"/>
                                    <xs:element ref="contacts:Suffix"/>
                                    <xs:element ref="contacts:Title"/>
                                    <xs:element ref="contacts:WebPage"/>
                                    <xs:element ref="contacts:YomiCompanyName"/>
                                    <xs:element ref="contacts:YomiFirstName"/>
                                    <xs:element ref="contacts:YomiLastName"/>
                                    <xs:element ref="contacts:Picture"/>
```



Copyright © 2010 Microsoft Corporation.

```
                        <xs:element ref="contacts2:CustomerId"/>
                        <xs:element ref="contacts2:GovernmentId"/>
                        <xs:element ref="contacts2:IMAddress"/>
                        <xs:element ref="contacts2:IMAddress2"/>
                        <xs:element ref="contacts2:IMAddress3"/>
                        <xs:element ref="contacts2:ManagerName"/>
                        <xs:element ref="contacts2:CompanyMainPhone"/>
                        <xs:element ref="contacts2:AccountName"/>
                        <xs:element ref="contacts2:NickName"/>
                        <xs:element ref="contacts2:MMS"/>
                        <xs:element ref="tasks:Complete"/>
                        <xs:element ref="tasks:Subject"/>
                        <xs:element ref="tasks:Categories"/>
                        <xs:element ref="tasks:DateCompleted"/>
                        <xs:element ref="tasks:DueDate"/>
                        <xs:element ref="tasks:UtcDueDate"/>
                        <xs:element ref="tasks:Importance"/>
                        <xs:element ref="tasks:Recurrence"/>
                        <xs:element ref="tasks:ReminderSet"/>
                        <xs:element ref="tasks:ReminderTime"/>
                        <xs:element ref="tasks:Sensitivity"/>
                        <xs:element ref="tasks:StartDate"/>
                        <xs:element ref="tasks:UtcStartDate"/>
                        <xs:element ref="notes:Subject"/>
                        <xs:element ref="notes:MessageClass"/>
                        <xs:element ref="notes:LastModifiedDate"/>
                        <xs:element ref="notes:Categories"/>
                        <xs:element ref="email:To"/>
                        <xs:element ref="email:From"/>
                        <xs:element ref="email:DateReceived"/>
                        <xs:element ref="email:InternetCPID"/>
                        <xs:element ref="email:Importance"/>
                        <xs:element ref="email:Flag"/>
                        <xs:element ref="email:Read"/>
                      </xs:choice>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Fetch">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="ServerId">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:maxLength value="64"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```



Copyright © 2010 Microsoft Corporation.

```
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Wait" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="59"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="HeartbeatInterval" minOccurs="0">
         <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="60"/>
            <xs:maxInclusive value="3540"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="WindowSize" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="512"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Partial" type="tns:EmptyTag" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

#### 2.2.2.19.1.1 Empty Sync Request

If the client sends a **Sync** request with no client additions, changes, or deletes, the server caches the request. If no changes are detected on the server, the **Sync** response includes only HTTP headers, and no XML payload, and is referred to as an empty **Sync** response.

When the client receives the empty **Sync** response, if there are no pending client changes, the client in turn can send only the HTTP headers, and no XML payload in the **Sync** request to save bandwidth. This request is referred to as an empty **Sync** request. If bandwidth is not a concern, the client can send a **Sync** request with an XML payload.

When the server receives the empty **Sync** request, the server assumes the request is identical to the cached request, so it retrieves the cached request and uses it. The cached request is discarded when the server receives a **Sync** request with an XML payload (a non-empty **Sync** request). This exchange of the empty **Sync** requests and responses continues until a change is detected on either the client or server. For an example empty **Sync** request and response, see section 4.5.10.

#### 2.2.2.19.1.2 Sync

The <Sync> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **Sync** command.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Collections><br><Partial> (request only)<br><Wait> (request only)<br><HeartbeatInterval> (request only)<br><WindowSize> (request only)<br><Limit> (response only)<br><Status> (response only) | **Container** | 1...1 (required) |

The <Limit> element and <Collections> element are mutually exclusive in a **Sync** response. That is, a **Sync** response can include either a <Limit> element or a <Collections> element, but not both.

#### 2.2.2.19.1.2.1 Collections

The <Collections> element serves as a container for the <Collection> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> | <Collection> | **Container** | 0...1 (optional) |

The <Collections> element appears both in synchronization requests and responses. The structure is identical.

The <Collections> element is optional. If <Collections> is present, it can contain multiple <Collection> elements.

Request/Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
<Collections>
    <Collection>
        ...
    </Collection>
</Collections>
```

##### 2.2.2.19.1.2.1.1 Collection

The <Collection> element wraps commands and options that apply to a particular collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collections> | <SyncKey><br><Supported> (request only)<br><CollectionId><br><DeletesAsMoves> (request only)<br><GetChanges> (request only)<br><WindowSize> (request only)<br><Options> (request only)<br><ConversationMode> (request only) | **Container** | 1...N (required) |



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| | <Status> (response only)<br><MoreAvailable> (response only)<br><Commands><br><Responses> (response only) | | |

The <Collection> element contains identification information (<CollectionID>), synchronization state (<SyncKey>), commands (<GetChanges>, <Commands>), and options (<WindowSize>, <Options>, <DeleteAsMoves>, <MoreAvailable>).

There is a strict ordering of the XML elements within a <Collection> node in a **Sync** request. The order is as follows:

1. <SyncKey>
2. <CollectionId>
3. <Supported>
4. <DeletesAsMoves>
5. <GetChanges>
6. <WindowSize>
7. <ConversationMode>
8. <Options>
9. <Commands>

The <Collection> element appears in both <Sync> requests and responses. The form is similar, although some child elements are valid in only one context.

A single <Collections> element can contain multiple <Collection> elements. Therefore, each <Collection> does not require its own **Sync** command. That is, a **Sync** request can specify multiple collections to be synchronized.

#### 2.2.2.19.1.2.1.1.1 SyncKey

The <SyncKey> element contains a value that is used by the server to mark the synchronization state of a collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | **String** (Up to 64 characters) | 1 (required) |

A synchronization key of value 0 initializes the synchronization state on the server and causes a full synchronization of the collection. The server sends a response that includes a new synchronization key value. The client MUST store this synchronization key value until the client requires the key value for the next synchronization request for that collection. When the client uses this synchronization key value to do the next synchronization of the collection, the client sends this synchronization key value to the server in a **Sync** request. If the synchronization is successful, the server responds by sending all objects in the collection. The response includes a new synchronization key value that the client MUST use on the next synchronization of the collection.



Copyright © 2010 Microsoft Corporation.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

#### 2.2.2.19.1.2.1.1.2 CollectionId

The <CollectionId> element specifies the server ID of the folder to be synchronized.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | **String** (Up to 64 characters) | 1…1 (required) |

The server ID of the folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

The <CollectionId> value "RI" specifies the recipient information cache, which is an information store that contains a list of contacts that the user has interacted with most often in the near term, and with whom the user is likely to interact again. The recipient information cache is a special folder (similar to the Inbox folder), which limits the operations that can be performed on it.

#### 2.2.2.19.1.2.1.1.3 Supported

The <Supported> element<67> is used by the client to specify which contact and calendar elements in a **Sync** request are managed by the client. Elements that are not named by the <Supported> element are said to be ghosted.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | See the description following this table. | **Container** | 0...1 (optional) |

By default, the server preserves the value of a ghosted element if that element is not included in a **Sync** request as a child of the <Supported> element. If an element is listed as a child of the <Supported> element, then the client is signifying that it will always transmit the current value of this element to the server. Once an element is listed as a <Supported> element, failure to provide a value for that element in a **Sync** request causes the server to delete the currently stored value.

The initial **Sync** request MUST include a <CollectionId> node, which MUST always precede the optional <Supported> node. See the <Collection> element (section 2.2.2.19.1.2.1.1) for the order of elements within the <Collection> node. This order is strictly enforced. A <Status> value of 4 is returned in the **Sync** response if the <CollectionId> element is not included in the **Sync** request. A <Status> value of 4 is also returned if the order of elements is incorrect within the <Collection> node.

```
<Collection>
   <SyncKey>0</SyncKey>
   <CollectionId>2</CollectionId>
   <Supported>
        <c:FirstName/>
        <c:MiddleName/>
        <c:LastName/>
        <c:HomePhoneNumber/>
        <c:MobilePhoneNumber/>
        <c:BusinessPhoneNumber/>
        <c:Email1Address/>
    </Supported>
</Collection>
```



Copyright © 2010 Microsoft Corporation.

The <Supported> element MUST be sent in a **Sync** command request when the <SyncKey> value (section 2.2.2.19.1.2.1.1.1) is set to zero. The server caches the list of <Supported> elements for subsequent synchronizations. If the <Supported> element is included when the <SyncKey> value is not set to zero, no error is returned, but the server ignores the request.

All of the Contact class [MS-ASCNTC] properties can be included as children of the <Supported> element.

To support elements of the Calendar class [MS-ASCAL], the following required elements MUST be included as children of the <Supported> element:

- <DtStamp> ([MS-ASCAL] section 2.2.2.7)
- <Categories> ([MS-ASCAL] section 2.2.2.17)
- <Sensitivity> ([MS-ASCAL] section 2.2.2.11)
- <BusyStatus> ([MS-ASCAL] section 2.2.2.4)
- <UID> ([MS-ASCAL] section 2.2.2.14)

The following elements are optional child elements of the <Supported> element when synchronizing the Calendar class.

- <Attendees> ([MS-ASCAL] section 2.2.2.16)
- <OrganizerName> ([MS-ASCAL] section 2.2.2.5)
- <OrganizerEmail> ([MS-ASCAL] section 2.2.2.6)
- <MeetingStatus> ([MS-ASCAL] section 2.2.2.15)
- <ResponseRequested> ([MS-ASCAL] section 2.2.2.20)<68>
- <DisallowNewTimeProposal> ([MS-ASCAL] section 2.2.2.23)<69>

For more information on which properties are ghosted by default, consult the ActiveSync class protocols, [MS-ASCAL] and [MS-ASCNTC].

#### 2.2.2.19.1.2.1.1.4 DeletesAsMoves

The <DeletesAsMoves> element indicates that any deleted items SHOULD be moved to the Deleted Items folder.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **Boolean** | 0...1 (optional) |

The <DeletesAsMoves> element appears only in requests to the server from the client. If the <DeleteAsMoves> element is set to false, the deletion is permanent.

If the client wants to permanently delete items, the request MUST include the <DeletesAsMoves> element with a value of 0 (FALSE). A value of 1 (TRUE), which is the default, indicates that any deleted items are moved to the Deleted Items folder. The default is assumed when the <DeletesAsMoves> element is either empty or not present.



Copyright © 2010 Microsoft Corporation.

The result of including the <DeletesAsMoves> element as a child of the <DeletesAsMoves> element is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.19.1.2.1.1.5 GetChanges

The <GetChanges> element requests the server to include in its response any pending changes to the collection that is specified by the <ServerId> element. If there have been changes since the last synchronization, the server response includes a <Commands> element that contains additions, deletions, and changes.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **Boolean** | 0...1 (optional) |

The result of including the <GetChanges> element as a child of the <GetChanges> element in a **Sync** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The <GetChanges> element appears only in requests to the server from the client.

If the client does not want the server changes returned, the request MUST include the <GetChanges> element with a value of 0 (FALSE). A value of 1 (TRUE), which is the default, indicates that the client wants the server changes to be returned. The default is assumed when the <GetChanges> element is either empty or not present.

A <Status> value of 4 is returned if <GetChanges> is set to True (1) when the <SyncKey> value is 0. No error is returned if the <GetChanges> element is set to False (0) when the <SyncKey> value is 0.

#### 2.2.2.19.1.2.1.1.6 WindowSize

The <WindowSize> element is sent from the client to the server to specify a maximum number of changed items in a collection or a request that SHOULD be included in the synchronization response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **Integer** | 0...1 (optional) |

The maximum value for the <WindowSize> element is 512. However, if the <WindowSize> element is set to 512, the server can send **Sync** response messages that contain less than 512 updates. If the server does not send all the updates in a single message, the **Sync** response message contains the <MoreAvailable> element, which indicates that there are additional updates on the server to be downloaded to the client.

The <WindowSize> element appears only in requests that are sent to the server from the client. If <WindowSize> is omitted, the server behaves as if a <WindowSize> element with a value of 100 were submitted. The server interprets values above 512 and 0 (zero) as 512.

<WindowSize> values less than 100 increase the load on the server, increase bandwidth, and decrease battery life because of the additional requests that are required to obtain all changes. <WindowSize> values larger than 100 cause larger responses, which are more susceptible to communication errors. A <WindowSize> value less than 100 can be useful if the client can display the initial set of objects while additional ones are still being retrieved from the server.



Copyright © 2010 Microsoft Corporation.

If the window size is changed during a synchronization transaction, the server returns a <MoreAvailable> element in the response. If this occurs, the client MUST synchronize again to continue getting items from the server.

The <WindowSize> element has been repurposed to also impose a global limit on the number of changes that are returned by the server. <WindowSize> can still be specified at the collection level and the server MUST honor both the global and collection level settings.

Collections are to be processed by the server in the order received, as follows:

- If the server has filled the <WindowSize> on a particular collection that has more changes, it will return the <MoreAvailable> element for that collection and continue to process the other collections until the global <WindowSize> has been filled.
- When the server has filled the global <WindowSize> and collections that have changes did not fit in the response, the server can return a <MoreAvailable> element.
- If a collection is not present in a **Sync** response, the client can assume that no changes are currently available for that collection.

The actual number of changes that are included in a **Sync** response for any particular collection depends on the <WindowSize> of the collection, the overall number of changes that are already included in the response, and the global <WindowSize>. The server will stop processing after the global <WindowSize> has been filled and simply not process the remaining collections. Any server-side changes that are pending in the unprocessed collections are picked up in the next synchronization.

The following synchronization request specifies that up to 100 changes be sent from the server back to the client. If there are more than 100 changes on the server, the <MoreAvailable> element is included in the response.

Request

```
<Collection>
      <Class>Email</Class>
      <SyncKey>1</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <WindowSize>100</WindowSize>
</Collection>
```

#### 2.2.2.19.1.2.1.1.7 ConversationMode

The optional element <ConversationMode><70> specifies whether to include items that are included within the conversation modality within the results of the **Sync** response. A single conversation MAY span multiple classes, and therefore <ConversationMode> is a child of the <Collection> element as opposed to the <Options> element.

Setting <ConversationMode> to **TRUE** results in retrieving all emails that match the conversations received within the date filter specified. However, although the body of the emails outside of that time based filter will not be retrieved, the text previews will be retrieved if the previews were requested.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | None | **Boolean** | 0...1 (optional) |

Setting the <ConversationMode> element to **FALSE** in a **Sync** request results in the synchronization of items that meet the criteria of the <FilterType> value. Setting <ConversationMode> to **TRUE** expands the result set to also include any items with identical <email2:ConversationId> values to those in the <FilterType> result set. The <ConversationMode> value has no impact on items outside the collection specified by the <CollectionId>, the result set is always limited to items in the specified collection. The <ConversationMode> value only limits or expands the results determined by the <FilterType> value.

The result of including the <ConversationMode> element as a child of a <ConversationMode> element is undefined. The server MAY return a protocol status error in response to such a command request.

Specifying <ConversationMode> for collections that do not store e-mails results in an invalid XML error, <Status> value 4.

#### 2.2.2.19.1.2.1.1.8 Options

The <Options> element is a container that encloses elements that control certain aspects of how the synchronization is performed.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (request only) | <FilterType><br><Conflict><br><MIMETruncation><br><MIMESupport><br><Class><br><MaxItems><br><airsyncbase:BodyPreference><br><airsyncbase:BodyPartPreference><br><rm:RightsManagementSupport> as specified in [MS-ASRM] section 2.2.2.1 | **Container** | 0...2 (optional) |

This element is optional, but if it is present, it SHOULD include at least one child element. The <Options> element appears only in requests to the server from the client.

Synchronization options enable the client to specify truncation and content settings. These settings are encapsulated within a <airsyncbase:BodyPreference> node within the <Options> element as follows:

```
<airsyncbase:BodyPreference>
    <airsyncbase:Type>1</airsyncbase:Type>
    <airsyncbase:TruncationSize>512</airsyncbase:TruncationSize>
    <airsyncbase:AllOrNone>1</airsyncbase:AllOrNone>
    <airsyncbase:Preview>255</airsyncbase:Preview>
</airsyncbase:BodyPreference>
```



Copyright © 2010 Microsoft Corporation.

Because synchronization options are specified on a collection, the client can specify a unique <airsyncbase:BodyPreference> for each collection that it is being synchronized. For more details about the <airsyncbase:BodyPreference> element, see [MS-ASAIRS] section 2.2.2.2.

The server preserves the <Options> block across requests, using a concept referred to as "sticky options". If the <Options> block is not included in a request, the previous <Options> block is used. Whenever the client specifies new options by including an <Options> block in the request, the server MUST replace the original <Options> block with the new <Options> block.

The following example <Options> element specifies that items in the collection that are older than three days SHOULD NOT be returned to the client, that items MUST be truncated to 512 characters if they are larger, and that, if there are any item conflicts, the server MUST replace the client items.

Request

```
<Collection>
    <Options>
        <FilterType>2</FilterType>
        <Conflict>1</Conflict>
        <MIMESupport>2</MIMESupport>
        <airsyncbase:BodyPreference>
           <airsyncbase:Type>4</airsyncbase:Type>
           <airsyncbase:TruncationSize>512</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
    </Options>
</Collection>
```

If two <Options> elements are included in a single **Sync** command request, one of the <Options> elements MUST specify the synchronization options for the SMS <Class>, while the other <Options> element specifies the options for the default <Class> of the given folder.

### 2.2.2.19.1.2.1.1.8.1 FilterType

The <FilterType> element specifies an optional time window for the objects that are sent from the server to the client. It applies to e-mail and calendar collections. If <FilterType> is specified, the server sends only objects that are dated within the specified time window.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Unsigned byte** | 0...1 (optional) |

The following table lists valid values for the element.

| Value | Meaning | Email? | Calendar? | Tasks? |
|---|---|---|---|---|
| 0 | No filter- synchronize all items | Yes | Yes | Yes |
| 1 | 1 day back | Yes | No | No |
| 2 | 3 days back | Yes | No | No |
| 3 | 1 week back | Yes | No | No |
| 4 | 2 weeks back | Yes | Yes | No |
| 5 | 1 month back | Yes | Yes | No |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Email? | Calendar? | Tasks? |
|---|---|---|---|---|
| 6 | 3 months back | No | Yes | No |
| 7 | 6 months back | No | Yes | No |
| 8 | Filter by incomplete tasks | No | No | Yes |

When the <FilterType> element is specified, the server manages objects on the client to maintain the time window. New objects are added when they are within the time window. The server sends <SoftDelete> operations for objects on the client when they become older than the window.

Calendar items that are in the future or that have recurrence but no end date are sent to the client regardless of the <FilterType> element value. Calendar items that fall between the <FilterType> value and the present time are returned to the client. For example, an appointment that occurred two weeks ago is returned when the <FilterType> value is set to 5 (1 month back). The result of including a <FilterType> value of 8 for a Calendar item is undefined. The server MAY return a protocol status error in response to such a command request.

The <FilterType> element is a child of the <Options> element. Therefore, it appears only in requests to the server from the client.

The result of including more than one <FilterType> element as the child of the <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

If the <FilterType> element is omitted, all objects are sent from the server without regard for their age. Clients MUST send a maximum of 1 <FilterType> element.

The server ignores the <FilterType> element if it is included in a contact **Sync** request, no error is thrown.

The server returns a <Status> value of 4 if a <FilterType> value of 8 is included in an email **Sync** request.

The following <Options> element in a synchronization request on an Inbox specifies that only e-mail messages that date back three days are returned to the client in the server synchronization response.

```
<Options>
    <FilterType>2</FilterType>
</Options>
```

### 2.2.2.19.1.2.1.1.8.2 Class

The optional element <Class><71> assigns the filters within the <Options> container to a given class.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **String** | 0...1 (optional) |

Options for the same <Class> within the same <Collection> MUST NOT be redefined. A <Status> value of 4 is returned if options for the same <Class> within the same <Collection> are redefined. The <Class> element is not necessary for the default items contained within the collection (contacts in a contacts folder for example).



Copyright © 2010 Microsoft Corporation.

For example, to sync SMS messages, include class "SMS". To also sync e-mail messages at the same time, include another <Options> node with class "Email".

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

The result of including more than one <Class> element as the child of the <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

### 2.2.2.19.1.2.1.1.8.3 Conflict

The <Conflict> element specifies how to resolve the conflict that occurs when an object has been changed on both the client and the server. The value specifies which object—the client object or the server object—to keep if there is a conflict.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Unsigned byte** | 0...1 (optional) |

The following table lists valid values for the element.

| Value | Meaning |
|---|---|
| 0 | Client object replaces server object. |
| 1 | Server object replaces client object. |

If the <Conflict> element is not present, the server object will replace the client object when a conflict occurs.

A value of 0 means to keep the client object; a value of 1 means to keep the server object. If the value is 1 and there is a conflict, a <Status> value of 7 is returned to inform the client that the object that the client sent to the server was discarded.

The <Conflict> element applies to the entire collection; therefore, it is not possible to apply the <Conflict> value to individual items within the collection.

The <Conflict> element is a child of the <Options> element, and therefore the <Conflict> element appears only in requests to the server from the client.

The result of including more than one <Conflict> element as the child of an <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

If a <Delete> element conflicts with an <Add> or <Change> element, the <Delete> takes precedence.



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.1.2.1.1.8.4 MIMESupport

The <MIMESupport> element is included in the <Options> element of a client **Sync** command request to enable MIME support for e-mail items that are sent from the server to the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Unsigned byte** | 0...1 (optional) |

The following table shows valid values for the element.

| Value | Meaning |
|---|---|
| 0 | Never send MIME data. |
| 1 | Send MIME data for S/MIME messages only. Send regular body for all other messages. |
| 2 | Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution. |

The client MUST send a maximum of one <MIMESupport> element. The result of including more than one <MIMESupport> element as the child of the <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

The **Sync** request MUST include the following in the <Options> element when handling S/MIME content:

- The <MIMESupport> element to tell the server to return MIME for S/MIME-only/All/None messages.
- The <airsyncbase:BodyPreference> element with its child element, <Type>, which contains a value of 4 to inform the server that the device can read the MIME BLOB.

When handling S/MIME content in the response, the server MUST include the <airsyncbase:Body> element, which is a child of the <ApplicationData> element. The <airsyncbase:Body> element is a complex element and MUST contain the following child nodes in an S/MIME synchronization response:

- The <airsyncbase:Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <airsyncbase:EstimatedDataSize> element to specify the rough total size of the data.
- The <airsyncbase:Truncated> element to indicate whether the MIME BLOB is truncated.
- The <airsyncbase:Data> element that contains the full MIME BLOB.

For more details about the <airsyncbase:Body> element or the <airsyncbase:BodyPreference> element, see [MS-ASAIRS] section 2.2.2.4 or 2.2.2.2, respectively.

### 2.2.2.19.1.2.1.1.8.5 MIMETruncation

The <MIMETruncation> element is included in the <Options> element of a client **Sync** command request to specify to the server whether the MIME data of the e-mail item SHOULD be truncated when it is sent from the server to the client.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Unsigned byte** | 0...1 (optional) |

The following table lists valid values for the element.

| Value | Meaning |
|---|---|
| 0 | Truncate all body text. |
| 1 | Truncate text over 4,096 characters. |
| 2 | Truncate text over 5,120 characters. |
| 3 | Truncate text over 7,168 characters. |
| 4 | Truncate text over 10,240 characters. |
| 5 | Truncate text over 20,480 characters. |
| 6 | Truncate text over 51,200 characters. |
| 7 | Truncate text over 102,400 characters. |
| 8 | Do not truncate; send complete MIME data. |

The size of the truncated message returned in the response is not the exact size of <MIMETruncation> value, the <MIMETruncation> size is an approximate value. This is because line feeds are treated as one character locally, but are counted as two characters during truncation.

The result of including more than one <MIMETruncation> element as the child of the <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.19.1.2.1.1.8.6 MaxItems

The optional element <MaxItems><72> specifies the maximum number of recipients to keep synchronized from within the recipient information cache. This element MUST only be included in a request when the <CollectionId> maps to the recipient information cache. The value of <MaxItems> does not specify the maximum estimate of additions and deletions to make to the recipient information cache, it only specifies the number of recipients to keep synchronized. A complete replacement of each recipient would be double the number of <MaxItems> or items in the store, as each recipient update requires a deletion and an addition.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Options> (request only) | None | **Integer** | 0...1 (optional) |

The result of including more than one <MaxItems> element as the child of the <Options> element is undefined. The server MAY return a protocol status error in response to such a command request.

#### 2.2.2.19.1.2.1.1.9 Commands

The <Commands> element is a container for operations that apply to a collection. Available operations are <Add>, <Delete>, <Change>, and <Fetch>. Client operations are sent in the **POST** request; server commands are sent in the **POST** response.



Copyright © 2010 Microsoft Corporation.

This element is optional. If it is present, it MUST include at least one operation. It is a child of the <Collection> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | <Add><br><Delete><br><Change><br><Fetch> (request only)<br><SoftDelete> (response only) | **Container** | 0...1 (optional) |

The result of including no child elements for the <Commands> element is undefined. The server MAY return a protocol status error in response to such a command request.

The <Commands> element can appear in both **Sync** requests and responses.

Request

```
<Collection>
<Commands>
 <Add>
 ...
 </Add>
 <Delete>
 ...
 </Delete >
 <Change>
 ...
 </Change >
 <Fetch>
 ...
 </Fetch>
 </Commands>
</Collection>
```

### 2.2.2.19.1.2.1.1.9.1 Change

The <Change> element modifies properties of an existing object on the client device or the server. The object to change is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands><br><Responses> (response only) | <ServerId><br><ApplicationData><br><Class> (response only)<br><Status> (response only) | **Container** | 0...N (optional) |

One or more <Change> elements can appear as a child of the <Commands> element for a particular collection.

Certain in-schema properties remain untouched in the following three cases:

- If there is only a <Flag>, <Read>, or <Categories> change (that is, if only a <Flag>, <Categories> or <Read> node is present), all other properties will remain unchanged and the



Copyright © 2010 Microsoft Corporation.

client SHOULD NOT send the other elements in the request. If all the other elements are sent, extra bandwidth is used, but no errors occur.

- If an <Exceptions> node is not specified, the properties for that exceptions node will remain unchanged. If an <Exception> node within the <Exceptions> node is not present, that particular exception will remain unchanged.
- If the <airsyncbase:Body>, <airsyncbase:Data>, or <contacts:Picture> elements are not present, the corresponding properties will remain unchanged.

In all other cases, if an in-schema property is not specified in a change request, the property is actively deleted from the item on the server. A client MUST be aware of this when it is sending **Sync** requests; otherwise, data can be unintentionally removed.

#### 2.2.2.19.1.2.1.1.9.1.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | None | **String** (Up to 64 characters) | 0…1 (optional) |

The client MUST store the server ID as an opaque string of up to 64 characters.

#### 2.2.2.19.1.2.1.1.9.1.2 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to change items on the client device or server. The format of this data is determined by the schema for the object.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASMS], and [MS-ASTASK]. | **Container** | 1 (required) |

#### 2.2.2.19.1.2.1.1.9.2 Delete

The <Delete> element deletes an object on the client or server. The object is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> | <ServerId><br><Class> (response only) | **Container** | 0...N (optional) |



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.1.2.1.1.9.2.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Delete> | None | **String** (Up to 64 characters) | 0…1 (optional) |

The client MUST store the server ID as an opaque string of up to 64 characters.

### 2.2.2.19.1.2.1.1.9.3 Add

The <Add> element can be used to create a new object in a collection on the client or on the server.

When a new item is being sent from the client to the server, the <ClientId> element specifies a temporary ID for the item, which is unique on the client. The <ApplicationData> element specifies the item data. The server then responds with an <Add> element in a <Responses> element, which specifies the client ID and the server ID that was assigned to the new item.

When the client sends a **Sync** command to the server and a new item has been added to the server collection since the last synchronization, the server responds with an <Add> element in a <Commands> element. This <Add> element specifies the server ID and data of the item to be added to the collection on the client.

When the client adds a calendar item, the <EndTime> element MUST be present in the <ApplicationData> element. A <Status> value of 6 is returned in the **Sync** response if the <EndTime> element is not included.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands><br><Responses> (response only) | <ServerId> (response only, see below)<br><ClientId><br><ApplicationData><br><Class><br><Status> (response only) | **Container** | 0...N (optional) |

One or more <Add> elements can appear as a child of the <Commands> and <Responses> elements for a particular collection.

The <Add> element cannot be used to add any e-mail items from the client to the server, or to modify the contents of the recipient information cache. If a client attempts to add e-mails to the server, or attempts to add items to the recipient information cache, then a <Status> value of 6 is returned.

If the server ID in an <Add> element from the server matches the server ID for an item on the client, the client treats the addition as a change to the client item.

The server is not required to send an individual response for every operation that is sent by the client. The client only receives responses for successful additions and fetches, and failed changes and deletions. When the client does not receive a response, the client SHOULD assume that the operation succeeded unless informed otherwise.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.19.1.2.1.1.9.3.1 Class

The optional element <Class><73> identifies the class of the item being added to the collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | None | **String** | 0…1 (optional) |

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

#### 2.2.2.19.1.2.1.1.9.3.2 ClientId

The <ClientId> is a unique identifier that is generated by the client to temporarily identify a new object that is being created by using the <Add> element. The client includes the <ClientId> element in the <Add> element request that it sends to the server. The server response contains an <Add> element that contains the original client ID and a new server ID that was assigned for the object, which replaces the client ID as the permanent object identifier.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (Request) | None | **String** (Typically an integer) | 1…1 (required) |

The <ClientId> element is a unique identifier that consists of up to 64 digits and letters. The client generates this ID. The value only has to be unique for the device during the duration of the **Sync** request that adds the object to the server. The client stores the client IDs until the synchronization session is completed successfully, to make recovery easier if the synchronization process fails.

An easy way to implement the client ID is to use a counter that is incremented for each new object that is created on the client.

#### 2.2.2.19.1.2.1.1.9.3.3 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to add items on the client device or server. The format of this data is determined by the schema for the object.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASMS], and [MS-ASTASK]. | **Container** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

The following <ApplicationData> element is used to add a contact item, identified by the <ServerId> element, to a folder on the client device.

Response

```
<Add>
          <ServerId> 2:6</ServerId>
          <ApplicationData>
          <A:Body></A:Body>
             <A:Email1Address>"jdobney@fourthcoffee.com"
&lt;jdobney@fourthcoffee.com&gt;</A:Email1Address>
                <A:FileAs>Dobney, JoLynn Julie</A:FileAs>
                <A:FirstName>JoLynn</A:FirstName>
                <A:HomePhoneNumber>425 555 1234</A:HomePhoneNumber>
                <A:MiddleName>Julie</A:MiddleName>
                <A:MobilePhoneNumber>425 555 1111</A:MobilePhoneNumber>
                <A:CompanyName>Fourth Coffee</A:CompanyName>
                <A:LastName>Dobney</A:LastName>
                <A:BusinessPhoneNumber>425 555 5555</A:BusinessPhoneNumber>
                <A:JobTitle>Usability Engineer</A:JobTitle>
          </ApplicationData>
</Add>
```

### 2.2.2.19.1.2.1.1.9.4 Fetch

The <Fetch> element is used to request the application data of an item that was truncated in a synchronization response from the server. The complete item is then returned to the client in a server response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> (request only)<br><Responses> (response only) | <ServerId><br><Status> (response only)<br><ApplicationData> (response only) | **Container** | 0...N (optional) |

The <Fetch> element cannot be used to get truncated calendar, contact, recipient information, or task items from the server.

### 2.2.2.19.1.2.1.1.9.4.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **String** (Up to 64 characters) | 0…1 (optional) |

The client MUST store the server ID as an opaque string of up to 64 characters.

### 2.2.2.19.1.2.2 Wait

The <Wait> element specifies, in a request, the number of minutes that the server SHOULD delay a response.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (request only) | None | **Integer** | 0...1 (optional) |

The result of including the <Wait> element as the child of the <Wait> element in a **Sync** command request is undefined. The server MAY return a protocol status error in response to such a command request.

Valid values for <Wait> are 1 through 59. When the client requests a wait-interval that is outside the acceptable range the server will send a response that includes a <Status> value of 14 and a <Limit> element.

Either <Wait> or <HeartbeatInterval> (section 2.2.2.19.1.2.3) can be specified, but not both. If both are specified, the server response will include a <Status> value of 4.

When <Wait> is used in a **Sync** request, the element indicates to the server that a response SHOULD be delayed either until the wait-interval, which is indicated by the contents of the <Wait> element, elapses or until any of the collections that are included in the request have changed.

It is at the discretion of the client to send the <Wait> element; the server is only guaranteed to respond immediately when <Wait> is not present. The client typically wants a server response immediately in the following cases:

- The client adds new items by using the <Add> element. In this case, an immediate response is required because the client requires the server-provided item ID to track changes to those new items.
- The client sends the server a <Change> element that contains an important update. In this case, a delayed response increases the possibility that the client has to resend the change because of a lost connection.

Although the server is only guaranteed to respond immediately when <Wait> and <HeartbeatInterval> are not present, the server SHOULD always respond immediately to a **Sync** request that includes an <Add> or a <Change>, unless the addition or change involves only flags.

A flagging change or a move out of (and not into) a folder which is being synced SHOULD NOT cause the request to finish early.

### 2.2.2.19.1.2.3 HeartbeatInterval

The <HeartbeatInterval> element specifies the number of seconds that the server SHOULD delay a response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (request only) | None | **Integer** | 0...1 (optional) |

The result of including the <HeartbeatInterval> element as the child of the <HeartbeatInterval> element in a **Sync** command request is undefined. The server MAY return a protocol status error in response to such a command request.

Valid values for <HeartbeatInterval> are 60 through 3540 seconds (59 minutes). When the client requests an interval that is outside the acceptable range the server will send a response that includes a <Status> value of 14 and a <Limit> element.



Copyright © 2010 Microsoft Corporation.

Either <HeartbeatInterval> or <Wait> (section 2.2.2.19.1.2.2) can be specified, but not both. If both are specified, the server response will include a <Status> value of 4.

When <HeartbeatInterval> is used in a **Sync** request, the element indicates to the server that a response SHOULD be delayed either until the interval, which is indicated by the contents of the <HeartbeatInterval> element, elapses or until any of the collections that are included in the request have changed.

It is at the discretion of the client to send the <HeartbeatInterval> element; the server is only guaranteed to respond immediately when neither <HeartbeatInterval> nor <Wait> (section 2.2.2.19.1.2.2) are present. The client typically requires a server response immediately in the following cases:

- The client adds new items by using the <Add> element. In this case, an immediate response is required because the client requires the server-provided item ID to track changes to those new items.
- The client sends the server a <Change> element that contains an important update. In this case, a delayed response increases the possibility that the client has to resend the change because of a lost connection.

Although the server is only guaranteed to respond immediately when <HeartbeatInterval> and <Wait> (section 2.2.2.19.1.2.2) are not present, the server SHOULD always respond immediately to a **Sync** request that includes an <Add> or a <Change>, unless the addition or change involves only flags.

A flagging change or a move out of (and not into) a folder which is being synced SHOULD NOT cause the request to finish early.

### 2.2.2.19.1.2.4 Partial

The <Partial> element indicates to the server that the client sent a partial list of collections, in which case the server obtains the rest of the collections from its cache.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (request only) | None | **Empty tag** | 0...1 (optional) |

The result of including the <Partial> element as the child of the <Partial> element in a **Sync** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The <Partial> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <Partial/> tag.

The client MUST NOT send a <Partial> element without any other elements in the **Sync** request. A **Sync** request is valid with just a <Partial> element and either a <Wait> or <HeartbeatInterval> element, a <WindowSize> element, a <Collections> element, or any combination of the three. A **Sync** request requires, at least, either a <Partial> element or a <Collections> element.

When a request includes a <Partial> element but does not specify some collections, the settings and synchronization key for each of those unspecified collections specified in the previous **Sync** request remain the same as specified in the previous request. Such a request is equivalent to a request that specifies each of these collections with the same settings and synchronization key as in the previous request. This enables the client to modify some aspect of the previous request (one of the



Copyright © 2010 Microsoft Corporation.

collections, the wait time, the global window size, and so on) without sending up every unchanged collection.

#### 2.2.2.19.1.2.5 WindowSize

The <WindowSize> element is sent from the client to the server to specify a maximum number of changed items in a collection or a request that SHOULD be included in the synchronization response.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (request only) | None | **Integer** | 0...1 (optional) |

The <WindowSize> element has been repurposed to also impose a global limit on the number of changes that are returned by the server. <WindowSize> can still be specified at the collection level and the server MUST honor both the global and collection level settings.

For details about the <WindowSize> element, see section 2.2.2.19.1.2.1.1.6.

### 2.2.2.19.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **Sync** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="AirSync:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="AirSync:"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:contacts="Contacts:"
xmlns:contacts2="Contacts2:" xmlns:calendar="Calendar:"
xmlns:email="Email:" xmlns:email2="Email2:" xmlns:airsyncbase="AirSyncBase:"
xmlns:tasks="Tasks:" xmlns:rm="RightsManagement:">
  <xs:import namespace="Contacts:" />
  <xs:import namespace="Contacts2:" />
  <xs:import namespace="Email:" />
  <xs:import namespace="Email2:" />
  <xs:import namespace="Calendar:" />
  <xs:import namespace="AirSyncBase:"/>
  <xs:import namespace="Tasks:"/>
  <xs:import namespace="RightsManagement:"/>

  <xs:complexType name="EmptyTag"/>
  <xs:element name="Sync">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:unsignedByte"/>
        <xs:choice>
          <xs:element minOccurs="0" name="Limit" type="xs:integer"/>
          <xs:element minOccurs="0" name="Collections">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Collection" minOccurs="0" maxOccurs="unbounded">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:choice maxOccurs="unbounded">
                        <xs:element name="SyncKey">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:maxLength value="64"/>
                            </xs:restriction>
```



Copyright © 2010 Microsoft Corporation.

```
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="CollectionId">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="Status" type="xs:unsignedByte"/>
                      <xs:element minOccurs="0" name="Commands">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element minOccurs="0" maxOccurs="unbounded" name="Delete">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Class" type="xs:string" minOccurs="0"
/>
                                  <xs:element name="ServerId">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:element>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element minOccurs="0" maxOccurs="unbounded"
name="SoftDelete">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="ServerId">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:element>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Class" type="xs:string" minOccurs="0"
/>
                                  <xs:element name="ServerId">
                                    <xs:simpleType>
                                      <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                      </xs:restriction>
                                    </xs:simpleType>
                                  </xs:element>
                                  <xs:element name="ApplicationData">
                                    <!--Data elements from the content classes are the
child elements of the ApplicaitonData eleent. For details about the content classes, see [MS-
ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK].-->
                                  </xs:element>
```



Copyright © 2010 Microsoft Corporation.

```
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="ServerId">
                                        <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
                                      <xs:element name="ApplicationData">
                                        <!--The data elements of the content classes and the
  AirSyncBase namespace are the child elements of the ApplicationData element. For details
  about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-
  ASTASK]. For details about the AirSyncBase namespace, see [MS-ASAIRS].-->
                                      </xs:element>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element minOccurs="0" name="Responses">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Class" type="xs:string" minOccurs="0"
  />
                                      <xs:element name="ServerId">
                                        <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
                                      <xs:element name="Status" type="xs:unsignedByte"/>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Class" type="xs:string" minOccurs="0"
  />
                                      <xs:element name="ClientId" type="xs:unsignedByte"/>
                                      <xs:element name="ServerId">
                                        <xs:simpleType>
                                          <xs:restriction base="xs:string">
                                            <xs:maxLength value="64"/>
                                          </xs:restriction>
                                        </xs:simpleType>
                                      </xs:element>
                                      <xs:element name="Status" type="xs:unsignedByte"/>
                                    </xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
                                </xs:complexType>
                              </xs:element>
                              <xs:element name="Fetch">
                                <xs:complexType>
                                  <xs:sequence>
                                    <xs:element name="ServerId">
                                      <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                          <xs:maxLength value="64"/>
                                        </xs:restriction>
                                      </xs:simpleType>
                                    </xs:element>
                                    <xs:element name="Status" type="xs:unsignedByte"/>
                                    <xs:element name="ApplicationData">
                                      <!--The data elements of the content classes and the
AirSyncBase namespace are the child elements of the ApplicationData element. For details
about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-
ASTASK]. For details about the AirSyncBase namespace, see [MS-ASAIRS].-->
                                      <xs:complexType>
                                        <xs:sequence>
                                          <xs:choice maxOccurs="unbounded">
                                            <xs:element ref="email:To"/>
                                            <xs:element ref="email:From"/>
                                            <xs:element ref="email:ReplyTo"/>
                                            <xs:element ref="email:Subject"/>
                                            <xs:element ref="email:DateReceived"/>
                                            <xs:element ref="email:DisplayTo"/>
                                            <xs:element ref="email:ThreadTopic"/>
                                            <xs:element ref="email:Importance"/>
                                            <xs:element ref="email:Read"/>
                                            <xs:element ref="airsyncbase:Attachments"/>
                                            <xs:element ref="airsyncbase:Body"/>
                                            <xs:element ref="email:MessageClass"/>
                                            <xs:element ref="email:MeetingRequest"/>
                                            <xs:element ref="email:InternetCPID"/>
                                            <xs:element ref="email:Flag"/>
                                            <xs:element ref="email:ContentClass"/>
                                            <xs:element ref="airsyncbase:NativeBodyType"/>
                                            <xs:element ref="rm:RightsManagementLicense"/>
                                          </xs:choice>
                                        </xs:sequence>
                                      </xs:complexType>
                                    </xs:element>
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                              <xs:element minOccurs="0" type="tns:EmptyTag" name="MoreAvailable"/>
                            </xs:choice>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:sequence>
```



Copyright © 2010 Microsoft Corporation.

```
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### 2.2.2.19.2.1 Sync

The <Sync> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **Sync** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <Collections><br><Partial> (request only)<br><Wait> (request only)<br><HeartbeatInterval> (request only)<br><WindowSize> (request only)<br><Limit> (response only)<br><Status> (response only) | **Container** | 0...1 (optional) |

The <Limit> element and <Collections> element are mutually exclusive in a **Sync** response. That is, a **Sync** response can include either a <Limit> element or a <Collections> element, but not both.

#### 2.2.2.19.2.1.1 Status

The <Status> element indicates the success or failure of the command. If the command failed, the <Status> element contains a code that indicates the type of failure.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (response only) | None | **Integer** | 1 (required) |

The following table lists the <Status> codes for the **Sync** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 3 | Invalid synchronization key. | Invalid or mismatched synchronization key.<br>—or—<br>Synchronization state corrupted on server. | Global | MUST return to <SyncKey> 0 for the collection. The client SHOULD either delete any items that were added since the last successful **Sync** or the client MUST add those items back to the server after completing the full resynchronization. |
| 4 | Protocol error. | There was a semantic error in the synchronization request. | Item or Global | The client is issuing a request that does not comply with the |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| | | | | specification requirements. Fix the request and retry. |
| 5 | Server error. | Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition. | Global | Retry the synchronization. If continued attempts to synchronization fail, consider returning to synchronization key 0. |
| 6 | Error in client/server conversion. | The client has sent a malformed or invalid item. | Item | Stop sending the item. This is not a transient condition. |
| 7 | Conflict matching the client and server object. | The client has changed an item for which the conflict policy indicates that the server's changes take precedence. | Item | If it is necessary, inform the user that the change they made to the item has been overwritten by a server change. |
| 8 | Object not found. | The client issued a <Fetch> or <Change> operation that has an ItemID value that is no longer valid on the server (for example, the item was deleted). | Item | Issue a synchronization request and prompt the user if necessary. |
| 9 | The **Sync** command cannot be completed. | User account could be out of disk space. | Item | Free space in the user's mailbox and try the **Sync** command again. |
| 12 | The folder hierarchy has changed. | Mailbox folders are not synchronized. | Item | Perform a **FolderSync** command and then retry the **Sync** command. |
| 13 | The **Sync** command request is not complete. | An empty or partial **Sync** command request is received and the cached set of notify-able collections is missing. | Item | Resend a full **Sync** command request. |
| 14 | Invalid <Wait> or <HeartbeatInterval> value. | The **Sync** request was processed successfully but the <Wait> or <HeartbeatInterval> interval that is specified by the client is outside the range set by the server administrator.<br>If the <HeartbeatInterval> or <Wait> value included in the **Sync** request is larger than the maximum allowable value, the response contains a <Limit> element that specifies the maximum allowed value.<br>If the <HeartbeatInterval> or <Wait> value included in the **Sync** request is smaller than the minimum allowable | Item | Update the <Wait> element value according to the <Limit> element and then resend the **Sync** command request. |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| | | value, the response contains a <Limit> element that specifies the minimum allowed value. | | |
| 15 | Invalid **Sync** command request. | Too many collections are included in the **Sync** request. | Item | Notify the user and synchronize fewer folders within one request. |
| 16 | Retry | Something on the server caused a retriable error. | Global | Resend the request. |

The <Status> element is sent only in responses from the server to the client.

#### 2.2.2.19.2.1.2 Limit

The <Limit> element specifies either the maximum number of collections that can be synchronized or the maximum/minimum value that is allowed for the <Wait> or <HeartbeatInterval> interval.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> (response only) | None | **Integer** | 0...1 (optional) |

The <Limit> element is returned in a response with a status code of 14 or 15. The value of the <Status> element indicates whether the limit applies to the <Wait> or <HeartbeatInterval> interval or the number of collections, as follows:

- A status code 14 indicates that <Limit> specifies the minimum or maximum wait-interval that is acceptable. When the value of the <Wait> or <HeartbeatInterval> element is outside the acceptable range, the server responds with the closest acceptable value. If a <Wait> value of less than 1 is sent, the server returns a <Limit> value of 1, indicating the minimum value of the <Wait> element is 1. If a <Wait> value greater than 59 is sent, the server returns a <Limit> value of 59, indicating the maximum value of <Wait> is 59. If a <HeartbeatInterval> value of less than 60 is sent, the server returns a <Limit> value of 60, indicating the minimum value of the <HeartbeatInterval> is 60. If a <HeartbeatInterval> value greater than 3540 is sent, the server returns a <Limit> value of 3540, indicating the maximum value of <HeartbeatInterval> is 3540.
- A status code 15 indicates that <Limit> specifies the maximum number of collections that can be synchronized.

#### 2.2.2.19.2.1.3 Collections

The <Collections> element serves as a container for the <Collection> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Sync> | <Collection> | **Container** | 0...1 (optional) |

The <Collections> element appears both in **Sync** requests and responses. The structure is identical.

The <Collections> element is optional. If <Collections> is present, it can contain multiple <Collection> elements, but MUST contain at least one.



Copyright © 2010 Microsoft Corporation.

Request/Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
<Collections>
    <Collection>
        ...
    </Collection>
</Collections>
```

### 2.2.2.19.2.1.3.1 Collection

The <Collection> element wraps operations and options that apply to a particular collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collections> | <SyncKey><br><Supported> (request only)<br><CollectionId><br><DeletesAsMoves> (request only)<br><GetChanges> (request only)<br><WindowSize> (request only)<br><Options> (request only)<br><Status> (response only)<br><MoreAvailable> (response only)<br><Commands><br><Responses> (response only) | **Container** | 1..512 (required) |

The <Collection> element contains identification information (<Class>, <CollectionID>), synchronization **state** (<SyncKey>), operations (<GetChanges>, <Commands>), and options (<WindowSize>, <Options>, <DeleteAsMoves>, <MoreAvailable>).

There is a strict ordering of the XML elements within a <Collection> node in a **Sync** request. The order is as follows:

- <SyncKey>
- <CollectionId>
- <Supported>
- <DeletesAsMoves>
- <GetChanges>
- <WindowSize>
- <Options>
- <Commands>

The <Collection> element appears in both **Sync** requests and responses. The form is similar, although some child elements are valid in only one context.



Copyright © 2010 Microsoft Corporation.

A single <Collections> element can contain multiple <Collection> elements. Therefore, each <Collection> does not require its own **Sync** command. That is, a **Sync** request can specify multiple <Collections> to be synchronized.

#### 2.2.2.19.2.1.3.1.1 SyncKey

The <SyncKey> element contains a value that is used by the server to represent the synchronization state of a collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | **String** (Up to 64 characters) | 1 (required) |

A synchronization key of value 0 initializes the synchronization state on the server and causes a full synchronization of the collection. The server sends a response that includes a new synchronization key value. The client MUST store this synchronization key value until the client requires the key value for the next synchronization request for that collection. When the client uses this synchronization key value to do the next synchronization of the collection, the client sends this synchronization key value to the server in a **Sync** request. If the synchronization is successful, the server responds by sending all objects in the collection. The response includes a new synchronization key value that the client uses on the next synchronization of the collection.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

The client MUST send a synchronization key value of 0 in an initial **Sync** request and the server sends a new synchronization key value in its response to the client. The client MUST NOT ignore the synchronization key value that is included in the initial response from the server.

#### 2.2.2.19.2.1.3.1.2 CollectionId

The <CollectionId> element specifies the server ID of the folder to be synchronized.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | None | <String> (Up to 64 characters) | 1 (required) |

The server ID of the folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

#### 2.2.2.19.2.1.3.1.3 Status

The <Status> element indicates the success or failure of the **Sync** command operation on the collection. If the operation failed, the <Status> element contains a code that indicates the type of failure.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| Collection | None | **Integer** | 1 (required) |

For details about the <Status> values returned by the **Sync** command, see section 2.2.2.19.2.1.1.



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.2.1.3.1.4 Commands

The <Commands> element is a container for operations that apply to a collection. Available operations are <Add>, <Delete>, <Change>, <Fetch>, and <SoftDelete>.

This element is optional. If it is present, it MUST include at least one operation. It is a child of the <Collection> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> | <Add><br><Delete><br><Change><br><Fetch> (request only)<br><SoftDelete> (response only) | **Container** | 0...1 (optional) |

The **Commands** element can appear in both **Sync** requests and responses.

Response

```
<Collection>
<Commands>
        <Add>
            …
        </Add>
        <Delete>
            …
        </Delete >
        <Change>
            …
        </Change >

        <SoftDelete>
            …
        </SoftDelete>
    </Commands>
</Collection>
```

### 2.2.2.19.2.1.3.1.4.1 Delete

The <Delete> element deletes an object on the client or server. The object is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> | <ServerId><br><Class> (response only)<74> | **Container** | 0...N (optional) |

### 2.2.2.19.2.1.3.1.4.1.1 Class

The optional <Class> element<75> identifies the class of the item being deleted from the collection.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Delete> (response only)<76> | None | **String** | 0...1 (optional) |

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

### 2.2.2.19.2.1.3.1.4.1.2 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Delete> | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the server ID as an opaque string of up to 64 characters.

### 2.2.2.19.2.1.3.1.4.2 SoftDelete

The <SoftDelete> element deletes an object from the client when it falls outside the <FilterType> results or it is no longer included as part of the **Sync** <Options> instructions. The object that is soft deleted is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> (response only) | <ServerId> (response only) | **Container** | 0...N (optional) |

The <SoftDelete> element contains any items that are filtered out of the **Sync** query due to being outside the <FilterType> date range, or no longer specified as part of the **Sync** <Options> instructions. For example, if the **Sync** <Options> element no longer specifies SMS class items as being synchronized, the current SMS items on the device will be soft deleted.

### 2.2.2.19.2.1.3.1.4.2.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <SoftDelete> | None | **String** (Up to 64 characters) | 1 (required) |



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.2.1.3.1.4.3 Change

The <Change> element contains properties of an existing object on the client device or the server that were modified. The object that changed is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> | <ServerId><br><ApplicationData><br><Class> (response only) | **Container** | 0...N (optional) |

One or more <Change> elements can appear as a child of the <Commands> element for a particular collection.

Certain in-schema properties remain untouched in the following three cases:

- If there is only a <Flag> or <Read> change (that is, if only a <Flag> or <Read> node is present), all other properties will remain unchanged and the server SHOULD NOT send the other elements. If the other elements are sent, extra bandwidth is used, but no errors occur.
- If an <Exceptions> node is not specified, the properties for that <Exceptions> node will remain unchanged. If an <Exception> node within the <Exceptions> node is not present, that particular <Exception> will remain unchanged.
- If <airsyncbase:Body>, <airsyncbase:Data>, or <contacts:Picture> nodes are not present, the corresponding properties will remain unchanged.

In all other cases, if an in-schema property is not specified in a change request, the property is actively deleted from the item on the server. A client MUST be aware of this when it is sending **Sync** requests; otherwise, data can be unintentionally removed.

#### 2.2.2.19.2.1.3.1.4.3.1 Class

The optional <Class> element<77> identifies the class of the item being changed in the collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> (response only) | None | **String** | 0...1 (optional) |

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

#### 2.2.2.19.2.1.3.1.4.3.2 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the server ID as an opaque string of up to 64 characters.

#### 2.2.2.19.2.1.3.1.4.3.3 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to change items on the client device or server. The format of this data is determined by the schema for the object.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]. | **Container** | 1 (required) |

### 2.2.2.19.2.1.3.1.4.4 Add

The <Add> element can be used to create a new object in a collection on the client or on the server.

When the client sends a **Sync** command to the server and a new item has been added to the server collection since the last synchronization, the server responds with an <Add> element in a <Commands> element. This <Add> element specifies the <ServerId> and data of the item to be added to the collection on the client.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> | <ServerId> (response only)<br><ApplicationData> | **Container** | 0...N (optional) |

One or more <Add> elements can appear as a child of the <Commands> element for a particular collection.

If the <ServerId> in an <Add> element from the server matches the <ServerId> for an item on the client, the client treats the addition as a change to the client item.

The server is not required to send an individual response for every operation that is sent by the client. The client only receives responses for successful additions and fetches, and failed changes and deletions. When the client does not receive a response, the client MUST assume that the operation succeeded unless informed otherwise.

#### 2.2.2.19.2.1.3.1.4.4.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **String** (Up to 64 characters) | 1 (required) |



Copyright © 2010 Microsoft Corporation.

The client MUST store the server ID as an opaque string of up to 64 characters.

#### 2.2.2.19.2.1.3.1.4.4.2 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to add items on the client device or server. The format of this data is determined by the schema for the object.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]. | **Container** | 1 (required) |

The following <ApplicationData> element is used to add a contact item, identified by the <ServerId> element, to a folder on the client device.

Response

```
<Add>
          <ServerId> 2:6</ServerId>
          <ApplicationData>
          <airsyncbase:Body>…</airsyncbase:Body>
             <contacts:Email1Address>"jdobney@fourthcoffee.com"
&lt;jdobney@fourthcoffee.com&gt;</contacts:Email1Address>
                <contacts:FileAs>Dobney, JoLynn Julie</contacts:FileAs>
                <contacts:FirstName>JoLynn</contacts:FirstName>
                <contacts:HomePhoneNumber>425 555 1234</contacts:HomePhoneNumber>
                <contacts:MiddleName>Julie</contacts:MiddleName>
                <contacts:MobilePhoneNumber>425 555 1111</contacts:MobilePhoneNumber>
                <contacts:CompanyName>Fourth Coffee</contacts:CompanyName>
                <contactsss:LastName>Dobney</contacts:LastName>
                <contacts:BusinessPhoneNumber>425 555 5555</contacts:BusinessPhoneNumber>
                <contacts:JobTitle>Usability Engineer</contacts:JobTitle>
          </ApplicationData>
</Add>
```

#### 2.2.2.19.2.1.3.1.4.5 Responses

The <Responses> element contains responses to operations that are processed by the server. Each response is wrapped in an element with the same name as the operation, such as <Add> and <Change>. The response contains a status code and other information, depending on the operation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (responses) | <Add>, <Fetch> (If the operation succeeded.)<br><Change> (If the operation failed.) | **Container** | 0...1 (optional) |

The <Responses> element appears only in responses that are sent from the server to the client. It is present only if the server has processed operation from the client. It is omitted otherwise (for



Copyright © 2010 Microsoft Corporation.

example, if the client requested server changes but had no changes to send to the server). If present, it MUST include at least one child element.

The following <Responses> element is part of a server response to a synchronization request. It shows items in the server collection that have been added, deleted, changed, or fetched.

Response

```
<Collection>
    <Responses>
        <Add>
            ...
        </Add>
        <Change>
            ...
        </Change>
        <Fetch>
            ...
        </Fetch>
    </Responses>
</Collection>
```

### 2.2.2.19.2.1.3.1.4.5.1 Change

The <Change> element contains properties of an existing object on the client device or the server that were modified. The object that changed is identified by its <ServerId> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Responses> (response only) | <ServerId><br><Class> (response only)<br><Status> (response only) | **Container** | 0...N (optional) |

### 2.2.2.19.2.1.3.1.4.5.1.1 Class

The optional <Class> element<78> identifies the class of the item being added to the collection.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> (response only) | None | **String** | 0...1 (optional) |

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.2.1.3.1.4.5.1.2 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the server ID as an opaque string of up to 64 characters.

### 2.2.2.19.2.1.3.1.4.5.1.3 Status

The <Status> element indicates the success or failure of the <Change> operation. If the operation failed, the <Status> element contains a code that indicates the type of failure.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Change> | None | **Integer** | 1 (required) |

For details about the <Status> values returned by the **Sync** command, see section 2.2.2.19.2.1.1.

### 2.2.2.19.2.1.3.1.4.5.2 Add

The <Add> element can be used to create a new object in a collection on the client or on the server.

When a new item is being sent from the client to the server, the <ClientId> element specifies a temporary ID for the item, which is unique on the client. The <ApplicationData> element specifies the item data. The server then responds with an <Add> element in a <Responses> element, which specifies the <ClientId> and the <ServerId> that was assigned to the new item.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Responses> (response only) | <ServerId> (response only)<br><ClientId><br><ApplicationData><br><Class> | **Container** | 0...N (optional) |

One or more <Add> elements can appear as a child of the <Responses> element for a particular collection.

If the <ServerId> in an <Add> element from the server matches the <ServerId> for an item on the client, the client treats the addition as a change to the client item.

The server is not required to send an individual response for every operation that is sent by the client. The client only receives responses for successful additions and fetches, and failed changes and deletions. When the client does not receive a response, the client MUST assume that the operation succeeded unless informed otherwise.

### 2.2.2.19.2.1.3.1.4.5.2.1 Class

The optional <Class> element<79> identifies the class of the item being changed in the collection.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | None | **String** | 0…1 (optional) |

The <Class> element is not included in **Sync** <Add> responses when the class of the collection matches the item class. For example, a **Sync** command response for an e-mail added to the Inbox folder would not include a <Class> element value of "Email" as the Inbox contains e-mail class content by default. However, a **Sync** command response for an SMS message added to the Inbox does include the <Class> value as SMS items are not the default class type of the Inbox.

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *SMS*

### 2.2.2.19.2.1.3.1.4.5.2.2 ClientId

The <ClientId> is a unique identifier that is generated by the client to temporarily identify a new object that is being created by using the <Add> element. The client includes the <ClientId> element in the <Add> element request that it sends to the server. The server response contains an <Add> element that contains the original client ID and a new server ID that was assigned for the object, which replaces the client ID as the permanent object identifier.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | None | **String** | Request: 1 (required)<br>Response: 1 (required) |

The <ClientId> element is a unique identifier that consists of up to 64 digits and letters. The client generates this ID. The value only has to be unique for the device during the duration of the **Sync** request that adds the object to the server. The client stores the client IDs until the synchronization session is completed successfully, to make recovery easier if the synchronization process fails.

An easy way to implement the client ID is to use a counter that is incremented for each new object that is created on the client.

### 2.2.2.19.2.1.3.1.4.5.2.3 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> (response only) | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the server ID as an opaque string of up to 64 characters.



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.19.2.1.3.1.4.5.2.4 Status

The <Status> element indicates the success or failure the <Add> operation. If the operation failed, the <Status> element contains a code that indicates the type of failure.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Add> | None | **Integer** | 1 (required) |

For details about the <Status> values returned by the **Sync** command, see section 2.2.2.19.2.1.1.

#### 2.2.2.19.2.1.3.1.4.5.3 Fetch

The <Fetch> element is used to request the application data of an item that was truncated in a synchronization response from the server. The complete item is then returned to the client in a server response.

The **ItemOperations** command (section 2.2.2.8) is the preferred way to fetch items.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Commands> (request only)<br><Responses> (response only) | <ServerId><br><Status> (response only)<br><ApplicationData> (response only) | **Container** | 0...N (optional) |

The <Fetch> element cannot be used to get truncated calendar, contact, or task items from the server.

#### 2.2.2.19.2.1.3.1.4.5.3.1 ServerId

The <ServerId> is a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **String** (Up to 64 characters) | 1 (required) |

The client MUST store the server ID as an opaque string of up to 64 characters.

#### 2.2.2.19.2.1.3.1.4.5.3.2 Status

The <Status> element indicates the success or failure of the <Fetch> operation. If the operation failed, the <Status> element contains a code that indicates the type of failure.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Fetch> | None | **Integer** | 1 (required) |

For details about the <Status> values returned by the **Sync** command, see section 2.2.2.19.2.1.1.



Copyright © 2010 Microsoft Corporation.

### 2.2.2.19.2.1.3.1.4.5.3.3 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to fetch items on the client device or server. The format of this data is determined by the schema for the object.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| Fetch | <airsyncbase:Body> as specified in [MS-ASAIRS] section 2.2.2.4<br><airsyncbase:BodyPart> as specified in [MS-ASAIRS] section 2.2.2.5<br><rm:RightsManagementLicense> as specified in [MS-ASRM] section 2.2.2.5<br>Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]. | **Container** | 1 (required) |

### 2.2.2.19.2.1.3.1.5 MoreAvailable

The <MoreAvailable> element is included in a synchronization response from the server to the client if there are more changes than the number that are requested in the <WindowSize> element.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Collection> (response only) | None | **Empty tag** | 0...1 (optional) |

The <MoreAvailable> element is an **Empty tag** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <MoreAvailable/> tag.

The <MoreAvailable> element appears only in responses that are sent from the server to the client. It appears only if the client request contained a <WindowSize> element and there are still changes to be returned to the client.

The server includes the <MoreAvailable> element in **Sync** responses that contain no additions, changes, or deletions when the server encounters elements external to the protocol. The client MUST continue to send **Sync** requests to retrieve additional changes until no additional results are sent by the server.<80>

The <MoreAvailable> element has no body. It is omitted if no additional changes are available. The maximum value for the <WindowSize> element is 512. The server interprets <WindowSize> values above 512 and 0 (zero) as 512.

If the <WindowSize> element is omitted, the server behaves as if a <WindowSize> element with a value of 100 was submitted. The <MoreAvailable> element is returned by the server if there are more than 512 changes, regardless of whether the <WindowSize> element is included in the request.

### 2.2.2.19.2.2 Empty Sync Response

The server sends a **Sync** response with only HTTP headers, and no XML payload, if there are no pending server changes to report to the client. This **Sync** response is referred to as an empty **Sync** response. The client can respond to the empty **Sync** response with an empty **Sync** request if there



Copyright © 2010 Microsoft Corporation.

are no pending client changes and if the client is required to conserve bandwidth; otherwise, a **Sync** request with an XML payload can be sent. For more information about empty **Sync** requests, see section 2.2.2.19.1.1. For an example empty **Sync** request and response, see section 4.5.10.

### 2.2.2.20 ValidateCert

The **ValidateCert** command is used by the client to validate a certificate that has been received via an S/MIME mail.

To validate a certificate, the server MUST verify that the certificate has not expired and has not been revoked. The server MUST walk up the certificate chain, verifying that each intermediate CA certificate has not expired and has not been revoked and that the root certificate is a trusted **certificate authority**. Certificate validation is particularly important for verifying signatures (for example, on S/MIME signed mail).

The ValidateCert namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the ValidateCert namespace use the namespace prefixes defined in section 2.2.1.

#### 2.2.2.20.1 Request

The following code shows the XSD [XMLSCHEMA1] for the **ValidateCert** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ValidateCert:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ValidateCert:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ValidateCert">
      <xs:complexType>
         <xs:all minOccurs="0">
            <xs:element name="CertificateChain" minOccurs="0">
               <xs:complexType>
                  <xs:choice maxOccurs="unbounded">
                     <xs:element name="Certificate" maxOccurs="unbounded">
                        <xs:simpleType>
                           <xs:restriction base="xs:base64Binary">
                              <xs:minLength value="4"/>
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="Certificates">
               <xs:complexType>
                  <xs:choice maxOccurs="unbounded">
                     <xs:element name="Certificate" maxOccurs="unbounded">
                        <xs:simpleType>
                           <xs:restriction base="xs:base64Binary">
                              <xs:minLength value="4"/>
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
            </xs:element>
            <xs:element name="CheckCrl" minOccurs="0">
```



Copyright © 2010 Microsoft Corporation.

```
                <xs:simpleType>
                   <xs:restriction base="xs:integer">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="1"/>
                   </xs:restriction>
                </xs:simpleType>
             </xs:element>
          </xs:all>
       </xs:complexType>
  </xs:element>
  </xs:schema>
```

#### 2.2.2.20.1.1 ValidateCert

The <ValidateCert> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **ValidateCert** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <CertificateChain> (request only)<br><Certificates> (request only)<br><CheckCRL> (request only)<br><Status> (response only)<br><Certificate> (response only) | **Container** | 1 (required) |

##### 2.2.2.20.1.1.1 CertificateChain

The <CertificateChain> element contains the list of certificates to be validated.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ValidateCert> (request only) | <Certificate> | **Container** | 0…1 (optional) |

###### 2.2.2.20.1.1.1.1 Certificate

The <Certificate> element contains the base64-encoded X509 certificate BLOB.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <CertificateChain> (request only)<br><ValidateCert> (response only) | <Status> (response only) | **String** (base64 encoded) | 1...N |

#### 2.2.2.20.1.1.2 Certificates

The <Certificates> element contains the list of certificates to be validated.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ValidateCert> (request only) | <Certificate> | **Container** | 1 (required) |



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.20.1.1.2.1 Certificate

The <Certificate> element contains the base64-encoded X509 certificate BLOB.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (request only)<br><ValidateCert> (response only) | <Status> (response only) | **String** (base64 encoded) | 1...N |

#### 2.2.2.20.1.1.3 CheckCRL

The <CheckCRL> element specifies whether the server SHOULD ignore an unverifiable revocation status.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ValidateCert> | None | **Integer** | 0...1 (optional) |

The revocation status of a certificate cannot be verified when the **certificate revocation lists (CRLs)** cannot be retrieved.

When <CheckCRL> is set to 1 (TRUE), the server MUST NOT ignore an unverifiable revocation status. When <CheckCRL> is set to 0 (FALSE), the server SHOULD ignore an unverifiable revocation status. The default value is 0.

### 2.2.2.20.2 Response

The following code shows the XSD [XMLSCHEMA1] for the **ValidateCert** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ValidateCert:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ValidateCert:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ValidateCert">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Status" type="xs:unsignedByte"/>
         <xs:element minOccurs="0" maxOccurs="unbounded" name="Certificate">
            <xs:complexType>
               <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="unbounded" name="Status"
type="xs:unsignedByte"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
</xs:schema>
```



Copyright © 2010 Microsoft Corporation.

#### 2.2.2.20.2.1 ValidateCert

The <ValidateCert> element is the top-level element in the XML stream. It identifies the body of the HTTP **POST** as containing a **ValidateCert** command.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| None | <CertificateChain> (request only)<br><Certificates> (request only)<br><CheckCRL> (request only)<br><Status> (response only)<br><Certificate> (response only) | **Container** | 1 (required) |

##### 2.2.2.20.2.1.1 Status

The <Status> element indicates whether one or more certificates were successfully validated.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ValidateCert> (response only) | None | **Integer** | 1...N (required) |

The following table lists the <Status> codes for the **ValidateCert** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | Protocol error. | Supplied protocol parameters are out of range or invalid. | Global | Fix client code. |
| 3 | The signature in the digital ID cannot be validated. | The signature in the certificate is invalid. | Item | Verify that the certificate has a valid signature. |
| 4 | The digital ID was issued by an untrusted source. | The certificate source is not trusted by the server. | Item | Contact the administrator to add the certificate to the trusted sources list if it is required. |
| 5 | The certificate chain that contains the digital ID was not created correctly. | Invalid, incorrectly formatted certificate. | Item | Verify that the certificate chain is formatted correctly. |
| 6 | The digital ID is not valid for signing e-mail messages. | The supplied certificate is not meant to be used for signing e-mail. | Item | Prompt the user. |
| 7 | The digital ID used to sign the message has expired or is not yet valid. | The certificate has expired. | Item | Obtain a new certificate. |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 8 | The time periods during which the digital IDs in the certificate chain are valid are not consistent. | One or more certificates in the chain could be out of date. | Item | Get the most recent certificate chain for the certificate. |
| 9 | A digital ID in the certificate chain is used incorrectly. | The supplied certificate is not valid for what it is being used for. | Item | Obtain a new certificate. |
| 10 | Information associated with the digital ID is missing or incorrect. | The certificate format is incorrect. | Item | Obtain a new certificate. |
| 11 | A digital ID in the certificate chain is used incorrectly. | A certificate that can only be used as an end-entity is being used as a certification authority (CA), or a CA that can only be used as an end-entity is being used as a certificate. | Item | Obtain the correct certificate chain. |
| 12 | The digital ID does not match the recipient's e-mail address. | Incorrect certificate was supplied, could be malicious. | Item | Obtain the correct certificate for the user. |
| 13 | The digital ID used to sign this message has been revoked. This can indicate that the issuer of the digital ID no longer trusts the sender, the digital ID was reported stolen, or the digital ID was compromised. | The certificate has been revoked by the certification authority that issued it. | Item | Obtain a new certificate. |
| 14 | The validity of the digital ID cannot be determined because the server that provides this information cannot be contacted. | The certificate revocation server is offline. | Item | Retry request after some time. |
| 15 | A digital ID in the chain has been revoked by the authority that issued it. | A certificate in the chain has been revoked. | Item | Obtain a new certificate. |
| 16 | The digital ID cannot be validated because its revocation status cannot be determined. | The signature in the certificate is invalid. | Item | Verify that the certificate has a valid signature. |
| 17 | An unknown server error has occurred. | The certificate source is not trusted by the server. | Item | Contact the administrator to add the certificate to the trusted sources list if it is necessary. |

#### 2.2.2.20.2.1.2 Certificate

The <Certificate> element contains the <Status> of the certificate validation.



Copyright © 2010 Microsoft Corporation.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <Certificates> (request only)<br><CertificateChain> (request only)<br><ValidateCert> (response only) | <Status> (response only) | **Container** | 0...N |

#### 2.2.2.20.2.1.2.1 Status

The <Status> element indicates whether one or more certificates were successfully validated.

| Parent elements | Child elements | Data type | Number allowed |
|---|---|---|---|
| <ValidateCert> (response only) | None | **Integer** | 1...N (required) |

The following table lists the <Status> codes for the **ValidateCert** command. For information about the scope of the <Status> value and for <Status> values common to all ActiveSync commands, see section 2.2.3.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| 1 | Success. | Server successfully completed command. | Global | None. |
| 2 | Protocol error. | Supplied protocol parameters are out of range or invalid. | Global | Fix client code. |
| 3 | The signature in the digital ID cannot be validated. | The signature in the certificate is invalid. | Item | Verify that the certificate has a valid signature. |
| 4 | The digital ID was issued by an untrusted source. | The certificate source is not trusted by the server. | Item | Contact the administrator to add the certificate to the trusted sources list if it is required. |
| 5 | The certificate chain that contains the digital ID was not created correctly. | Invalid, incorrectly formatted certificate. | Item | Verify that the certificate chain is formatted correctly. |
| 6 | The digital ID is not valid for signing e-mail messages. | The supplied certificate is not meant to be used for signing e-mail. | Item | Prompt the user. |
| 7 | The digital ID used to sign the message has expired or is not yet valid. | The certificate has expired. | Item | Obtain a new certificate. |
| 8 | The time periods during which the digital IDs in the certificate chain are valid are not consistent. | One or more certificates in the chain could be out of date. | Item | Get the most recent certificate chain for the certificate. |
| 9 | A digital ID in the certificate chain is used incorrectly. | The supplied certificate is not valid for what it is | Item | Obtain a new certificate. |



Copyright © 2010 Microsoft Corporation.

| Value | Meaning | Cause | Scope | Resolution |
|---|---|---|---|---|
| | | being used for. | | |
| 10 | Information associated with the digital ID is missing or incorrect. | The certificate format is incorrect. | Item | Obtain a new certificate. |
| 11 | A digital ID in the certificate chain is used incorrectly. | A certificate that can only be used as an end-entity is being used as a certification authority (CA), or a CA that can only be used as an end-entity is being used as a certificate. | Item | Obtain the correct certificate chain. |
| 12 | The digital ID does not match the recipient's e-mail address. | Incorrect certificate was supplied, could be malicious. | Item | Obtain the correct certificate for the user. |
| 13 | The digital ID used to sign this message has been revoked. This can indicate that the issuer of the digital ID no longer trusts the sender, the digital ID was reported stolen, or the digital ID was compromised. | The certificate has been revoked by the certification authority that issued it. | Item | Obtain a new certificate. |
| 14 | The validity of the digital ID cannot be determined because the server that provides this information cannot be contacted. | The certificate revocation server is offline. | Item | Retry request after some time. |
| 15 | A digital ID in the chain has been revoked by the authority that issued it. | A certificate in the chain has been revoked. | Item | Obtain a new certificate. |
| 16 | The digital ID cannot be validated because its revocation status cannot be determined. | The signature in the certificate is invalid. | Item | Verify that the certificate has a valid signature. |
| 17 | An unknown server error has occurred. | The certificate source is not trusted by the server. | Item | Contact the administrator to add the certificate to the trusted sources list if it is necessary. |

### 2.2.3 Common Status Codes

The <Status> values returned for each command are specified in the <Status> section for each command. Links to each Status section are listed in the following table.

| Command | Status value section |
|---|---|
| Autodiscover | Section 2.2.2.1.2.1.1.3.3.1 |
| FolderCreate | Section 2.2.2.2.2.1.1 |



Copyright © 2010 Microsoft Corporation.

| Command | Status value section |
|---|---|
| FolderDelete | Section 2.2.2.3.2.1.1 |
| FolderSync | Section 2.2.2.4.2.1.1 |
| FolderUpdate | Section 2.2.2.5.2.1.1 |
| GetItemEstimate | Section 2.2.2.7.2.1.1.1 |
| ItemOperations | Section 2.2.2.8.3.1.1 |
| MeetingResponse | Section 2.2.2.9.2.1.1.2 |
| MoveItems | Section 2.2.2.10.2.1.1.2 |
| Ping | Section 2.2.2.11.2.1.1 |
| Provision | [MS-ASPROV] section 3.1.5.2 |
| ResolveRecipients | Section 2.2.2.13.2.1.1 |
| Search | Section 2.2.2.14.2.1.1 |
| SendMail | Section 2.2.2.15.2.1.1 |
| Settings | Section 2.2.2.16.2.1.1 |
| SmartForward | Section 2.2.2.17.2.1.1 |
| SmartReply | Section 2.2.2.18.2.1.1 |
| Sync | Section 2.2.2.19.2.1.1 |
| ValidateCert | Section 2.2.2.20.2.1.1 |

Many of the <Status> codes listed in the command sections have a scope assigned to them. The following table defines the scope values.

| Scope Value | Description |
|---|---|
| Global | The status pertains to the overall client request. |
| Item | The status pertains to a particular item within the overall client request. |
| Policy | The status pertains to a particular policy within the **Provision** command. |

In addition to the values specified for individual commands, the following <Status> values are common to all commands. They are used in addition to the specific error codes described with the previous sections of this document.

| Value | Name | Description |
|---|---|---|
| 101 | InvalidContent | The body of the HTTP request sent by the client is invalid.<81><br>Ensure the HTTP request is using the specified Content-Type and length, and that the request is not missing (when |



Copyright © 2010 Microsoft Corporation.

| Value | Name | Description |
|---|---|---|
| | | an empty body is not allowed).<br>Examples:<br>**Ping** with a text/plain body, or **SendMail** with version 12.1 and a WBXML body. |
| 102 | InvalidWBXML | The request contains WBXML but it could not be decoded into XML. |
| 103 | InvalidXML | The XML provided in the request did not follow the protocol requirements. |
| 104 | InvalidDateTime | The request contains a timestamp that could not be parsed into a valid date and time. |
| 105 | InvalidCombinationOfIDs | The request contained a combination of parameters that is invalid. |
| 106 | InvalidIDs | The request contains one or more IDs that could not be parsed into valid values.<br>That is different from specifying an ID in the proper format that does not resolve to an existing item.<82> |
| 107 | InvalidMIME | The request contains a MIME that could not be parsed. |
| 108 | DeviceIdMissingOrInvalid | The DeviceID is either missing or has an invalid format. |
| 109 | DeviceTypeMissingOrInvalid | The DeviceType is either missing or has an invalid format. |
| 110 | ServerError | The server encountered an unknown error, the device should not retry later.<83> |
| 111 | ServerErrorRetryLater | The server encountered an unknown error, the device should retry later.<84> |
| 112 | ActiveDirectoryAccessDenied | The server does not have access to read/write to an object in the directory service.<85> |
| 113 | MailboxQuotaExceeded | The mailbox has reached its size quota.<86> |
| 114 | MailboxServerOffline | The mailbox server is offline. |
| 115 | SendQuotaExceeded | The request would exceed the "send" quota. |
| 116 | MessageRecipientUnresolved | One of the recipients could not be resolved to an e-mail address. |



Copyright © 2010 Microsoft Corporation.

| Value | Name | Description |
|---|---|---|
| 117 | MessageReplyNotAllowed | The mailbox server will not allow a reply of this message. |
| 118 | Message PreviouslySent | The message was already sent in a previous request. The server determined this by remembering the <ClientId> values of the last few sent messages. This request contains a <ClientId> that was already used in a recent message. |
| 119 | MessageHasNoRecipient | The message being sent contains no recipient. |
| 120 | MailSubmissionFailed | The server failed to submit the message for delivery. |
| 121 | MessageReplyFailed | The server failed to create a reply message. |
| 122 | AttachmentIsTooLarge | The attachment is too large to be processed by this request. |
| 123 | UserHasNoMailbox | A mailbox could not be found for the user. |
| 124 | UserCannotBeAnonymous | The request was sent without credentials. Anonymous requests are not allowed. |
| 125 | UserPrincipalCouldNotBeFound | The user was not found in the directory service. |
| 126 | UserDisabledForSync | The user object in the directory service indicates that this user is not allowed to use ActiveSync. |
| 127 | UserOnNewMailboxCannotSync | The server is configured to prevent users from syncing. |
| 128 | UserOnLegacyMailboxCannotSync | The server is configured to prevent users on legacy servers from syncing. |
| 129 | DeviceIsBlockedForThisUser | The user is configured to allow only some devices to sync. This device is not the allowed device. |
| 130 | AccessDenied | The user is not allowed to perform that request. |
| 131 | AccountDisabled | The user's account is disabled. |
| 132 | SyncStateNotFound | The sync state for the device was unexpectedly missing. It might have disappeared while the request was in progress. The next request will likely answer a sync key error and the device will be forced to do full sync.<87> |



Copyright © 2010 Microsoft Corporation.

| Value | Name | Description |
|---|---|---|
| 133 | SyncStateLocked | The sync state is locked. Possibly because the mailbox is being moved or was recently moved. |
| 134 | SyncStateCorrupt | The sync state appears to be corrupt. |
| 135 | SyncStateAlreadyExists | The sync state for this device already exists. This can happen with two initial syncs are executed concurrently. |
| 136 | SyncStateVersionInvalid | The sync state version is invalid. |
| 137 | CommandNotSupported | The command is not supported by this server.<88> |
| 138 | VersionNotSupported | The command is not supported in the protocol version specified.<89> |
| 139 | DeviceNotFullyProvisionable | The device uses a protocol version that cannot send all the policy settings the admin enabled. |
| 140 | RemoteWipeRequested | A remote wipe was requested. The device should provision to get the request and then do another provision to acknowledge it.<90> |
| 141 | LegacyDeviceOnStrictPolicy | A policy is in place but the device is not provisionable. |
| 142 | DeviceNotProvisioned | There is a policy in place, the device needs to provision.<91> |
| 143 | PolicyRefresh | The policy is configured to be refreshed every few hours. The device needs to re-provision. |
| 144 | InvalidPolicyKey | The device's policy key is invalid. The policy has probably changed on the server. The device needs to re-provision. |
| 145 | ExternallyManagedDevicesNotAllowed | The device claimed to be externally managed, but the server doesn't allow externally managed devices to sync. |
| 146 | NoRecurrenceInCalendar | The request tried to forward an occurrence of a meeting that has no recurrence. |
| 147 | UnexpectedItemClass<92> | The request tried to operate on a type of items unknown to the server. |
| 148 | RemoteServerHasNoSSL | The request needs to be proxied to another server but that server doesn't have Secure Sockets Layer (SSL) enabled. This server is configured to only proxy requests to servers with SSL |



Copyright © 2010 Microsoft Corporation.

| Value | Name | Description |
|---|---|---|
| | | enabled. |
| 149 | InvalidStoredRequest | The server had stored the previous request from that device. When the device sent an empty request, the server tried to re-execute that previous request but it was found to be impossible. The device needs to send the full request again. |
| 150 | ItemNotFound | The <ItemId> specified in the **SmartReply** or **SmartForward** request could not be found in the mailbox. |
| 151 | TooManyFolders | The mailbox contains too many folders. By default, the mailbox cannot contain more than 1000 folders. |
| 152 | NoFoldersFound | The mailbox contains no folders. |
| 153 | ItemsLostAfterMove | After moving items to the destination folder, some of those items could not be found. |
| 154 | FailureInMoveOperation | The mailbox server returned an unknown error while moving items. |
| 155 | MoveCommandDisallowedForNonPersistentMoveAction | An **ItemOperations** request to move a conversation is missing the <MoveAlways> element. |
| 156 | MoveCommandInvalidDestinationFolder | The destination folder for the move is invalid. |
| 166 | InvalidAccountId | The <AccountId> value is not valid.<93> |
| 167 | AccountSendDisabled | The <AccountId> value specified in the request does not support sending e-mail.<94> |
| 168 | IRM_FeatureDisabled | The Information Rights Management feature is disabled.<95> |
| 169 | IRM_TransientError | Information Rights Management encountered an error.<96> |
| 170 | IRM_PermanentError | Information Rights Management encountered an error.<97> |
| 171 | IRM_InvalidTemplateID | The <TemplateID> value is not valid.<98> |
| 172 | IRM_OperationNotPermitted | Information Rights Management does not support the specified operation.<99> |



Copyright © 2010 Microsoft Corporation.

| Value | Name | Description |
|---|---|---|
| 173 | NoPicture | The user does not have a contact photo.<100> |
| 174 | PictureTooLarge | The contact photo exceeded the size limit set by the <MaxSize> element.<101> |
| 175 | PictureLimitReached | The number of contact photos returned exceeded the size limit set by the <MaxPictures> element.<102> |
| 176 | BodyPart_ConversationTooLarge | The conversation is too large to compute the body parts. Try requesting the body of the item again, without body parts.<103> |
| 177 | MaximumDevicesReached | The user's account has too many device partnerships. Delete partnerships on the server before proceeding.<104> |



Copyright © 2010 Microsoft Corporation.

# 3 Protocol Details

## 3.1 Common 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

The client creates request messages consisting of an HTTP header, as specified in [MS-ASHTTP], and the XML command to be performed on the server, as specified in section 2.2.1. The request message is sent to the server by the client and a response message is received back from the server.

#### 3.1.5.1 Downloading Policy Settings

This section describes how the client device can download policy settings from the server by using the **Provision** command.

The first command the client issues to the server SHOULD be a **Provision** command, as specified in [MS-ASPROV]. The client can send the HTTP **OPTIONS** command to the server before sending the **Provision** command, to retrieve server settings, but the HTTP **OPTIONS** command is optional. Sending other commands to the server before the **Provision** command results in a <Status> value of 142 being returned to the client.

The initial **Provision** command request MUST contain the <provision:PolicyType> element, which specifies the format in which the policy settings are provided. If the <provision:PolicyType> element is not included in the initial **Provision** command request, the server responds with a <provision:Status> value of 2. The server then responds with the <provision:PolicyType>, <provision:PolicyKey>, and <provision:Data> elements. The <provision:PolicyKey> is used by the server to mark the state of policy settings on the client device. The policy settings, in the format specified in the <provision:PolicyType> element, are contained in the <provision:Data> element.

The client device then applies the policy settings that were received from the server and sends an acknowledgement back to the server in another **Provision** command request. The acknowledgement from the client device contains <provision:PolicyType>, <provision:PolicyKey>,



Copyright © 2010 Microsoft Corporation.

and <provision:Status> elements. The <provision:Status> element indicates whether the policy settings were successfully applied by the client. The response from the server contains <provision:PolicyType>, <provision:PolicyKey>, and <provision:Status> elements. The <provision:Status> element indicates whether the server successfully recorded the client's acknowledgement.

The following figure shows the process for downloading policy settings.

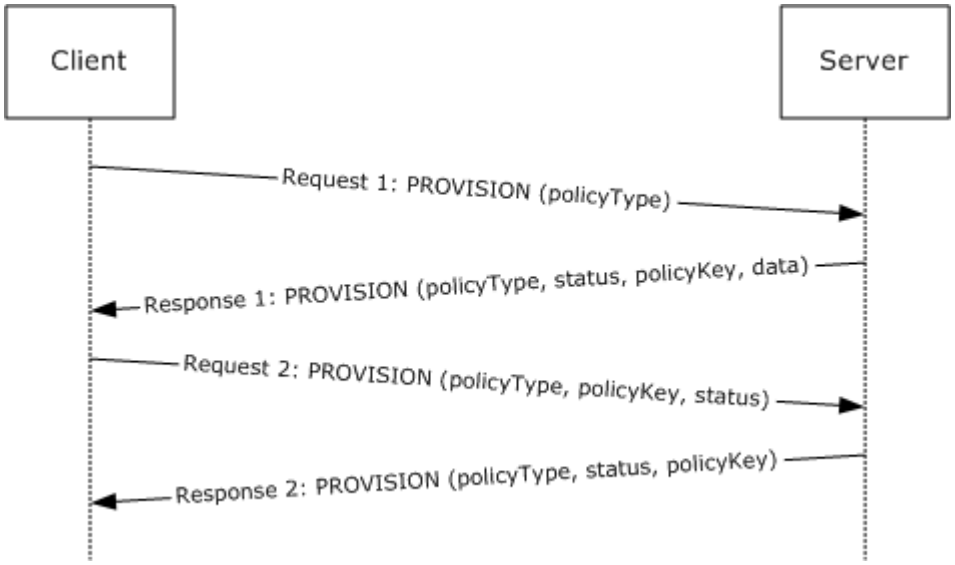


**Figure 4: Downloading policy settings**

The following table lists the command sequence for downloading policy settings.

| Order | Client action | Server action |
|---|---|---|
| 1 | The client sends a **Provision** command request with the type of policy settings to be downloaded. | The server response contains the policy type, policy key, data, and status code. |
| 2 | The client acknowledges that it received and applied the policy settings by sending another **Provision** command request with the policy type, policy key, and status code. | The server response contains the policy type, policy key, and status code to indicate that the server recorded the client's acknowledgement. |

### 3.1.5.2 Setting Device Information

This section describes how to use the **Settings** command to set device information on the server.

Clients SHOULD send <settings:DeviceInformation> parameters to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command, so that the server can use this information to determine what the device has access to.<105>

The client sets device information by sending an initial **Settings** command request to the server with the <settings:Set> element identifying <settings:DeviceInformation> parameters as specified in section 2.2.2.16.1.1.4.



Copyright © 2010 Microsoft Corporation.

### 3.1.5.3 Synchronizing a Folder Hierarchy

This section describes how to use the **FolderSync** command to replicate the folder hierarchy of the user's mailbox on the client.

The client initiates folder synchronization by sending an initial **FolderSync** command request to the server with a <folderhierarchy:SyncKey> key of zero (0). The server responds with a new <folderhierarchy:SyncKey> value and provides a list of all the folders in the user's mailbox. The folders are identified by a <folderhierarchy:ServerId>, which can then be used in a **Sync** command to synchronize the items in those folders.

Additional folder synchronizations can be performed by using the <folderhierarchy:SyncKey> from the initial **FolderSync** command response to get folder additions, deletions, or updates from the server. At any point, the client can repeat the initial **FolderSync** command, sending a <SyncKey> of zero (0), and resynchronizing the entire hierarchy. Existing <folderhierarchy:ServerId> values do not change when the client resynchronizes.

The client can use the **GetItemEstimate** command to obtain an estimate of the number of items that need to be synchronized in a collection, which is useful when the client UI displays a progress bar while it retrieves items from the server. The client can also limit the number of changed items returned in the **Sync** response by submitting the <airsync:WindowSize> element, which specifies the maximum number of items to synchronize at one time. If the number of items returned is larger than the <airsync:WindowSize>, the <airsync:MoreAvailable> element is returned in the **Sync** command response. The client then continues to call the **Sync** command until no more items are available.

The following table lists the command sequence for folder hierarchy synchronization.

The asterisk (*) in the Order column means that a step is run once and can be repeated multiple times.

| Order | Client action | Server action |
|---|---|---|
| 1 | The client sends the **FolderSync** command with the <folderhierarchy:SyncKey> element set to zero (0) to get the folder hierarchy and the <folderhierarchy:ServerId> values of all the folders. | The server response contains the folder hierarchy and a new <folderhierarchy:SyncKey>. The client stores the names and <folderhierarchy:ServerId> values of all folders that can be synchronized. |
| 2* | The client sends the **FolderSync** command with the new <folderhierarchy:SyncKey> value to update the folder hierarchy. | If any changes have occurred on the server, the new, deleted, or changed folders are returned to the client. |

The folder hierarchy is now populated on the client and ready for the contents of the folders to be synchronized.

### 3.1.5.4 Synchronizing Inbox, Calendar, Contacts, and Tasks Folders

The client synchronizes the contents of individual folders by using the **Sync** command. The client can synchronize the Inbox, Calendar, or Contacts folder, or any folder within the mailbox after the folder hierarchy has been populated by the **FolderSync** command, as described in section 3.1.5.3.

In order to synchronize the content of each of the folders, an initial synchronization key for each folder MUST be obtained from the server. The client obtains the key by sending the server an initial **Sync** request with a <airsync:SyncKey> value of zero (0) and the <airsync:CollectionId> that



Copyright © 2010 Microsoft Corporation.

identifies the folder to be synchronized. The **Sync** command response includes a new <airsync:SyncKey> value, which is generated by the server for each transaction.

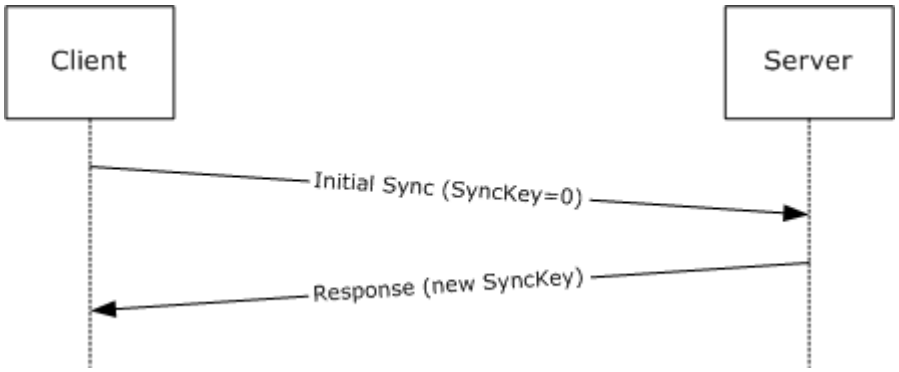


**Figure 5: Retrieval of SyncKey value**

The <airsync:SyncKey> issued in the initial **Sync** response MUST be stored by the client, and is sent in the second **Sync** request. The second **Sync** request includes the new <airsync:SyncKey> element as well as the <airsync:GetChanges> element.

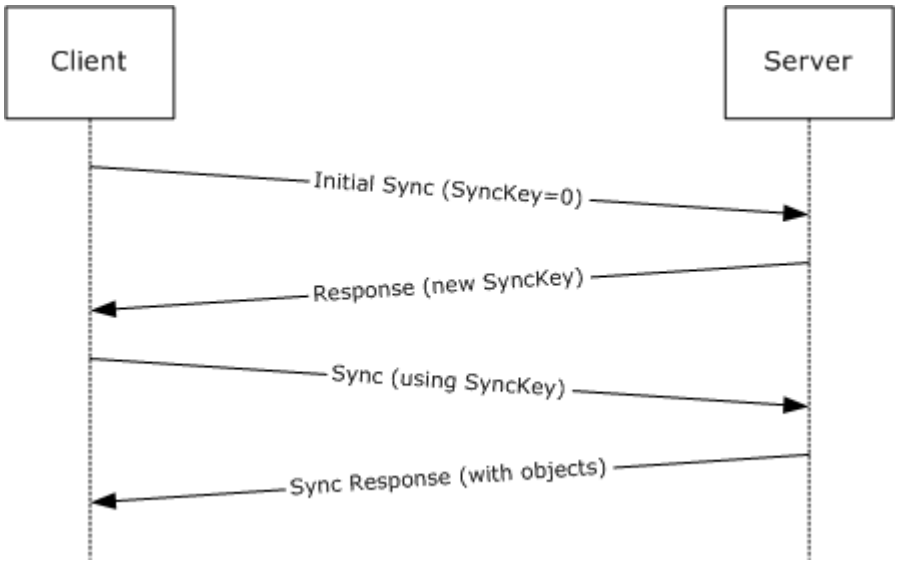


**Figure 6: Retrieval of folder content**

The server responds by adding all the items in the collection to the client and returning a new <airsync:SyncKey>, which can be used in successive synchronizations. The client deletes its copy of all objects in the collection that are being synchronized before the client performs a full synchronization. The client can use the **GetItemEstimate** command to obtain an estimate of the number of items that have to be synchronized before completely synchronizing a collection, which is useful when the client user interface (UI) displays a progress bar while getting items from the server. In some cases, the client could have to submit a <airsync:WindowSize> element that specifies the number of items to be synchronized at a time.

If more items remain to be synchronized, the <airsync:MoreAvailable> element is returned in the **Sync** command response. The client then continues to call the **Sync** command until no more items



Copyright © 2010 Microsoft Corporation.

are available. For more details about the <airsync:WindowSize> element of the **Sync** command, see section 2.2.2.19.1.2.1.1.6. The following figure shows the folder synchronization process.

After a full synchronization has been performed on a collection, successive synchronizations are used to obtain additions, deletions, or changes to the initial collection state. The client can use the **Sync** command request to add, delete, or change items on the server, and the server can use the **Sync** command response to add, delete, or change items on the client.

The following table lists the command sequence for folder synchronization.

The asterisk (*) in the Order column means that the step can be repeated multiple times. [n] means that a step is optional.

| Order | Client action | Server action |
|---|---|---|
| 1 | The client sends the **Sync** command for the Email, Calendar, Contacts, and/or Tasks collection with a synchronization key of zero (0). This establishes a partnership with the server, initializing server data for the device. | The server response contains the synchronization key for the collection, to be used in successive synchronizations. |
| 2* | The client sends the **Sync** command with a synchronization key of zero (0) for other collections to be synchronized. | The server responds with new synchronization keys for each collection. |
| [3] | The client sends the **GetItemEstimate** command for all collections to be synchronized. This step can be skipped if it is not required by the client UI. | The server response indicates how many items will be added, changed, or deleted, for each collection. |
| 4* | The client sends the **Sync** command with the <airsync:GetChanges> element for a collection. The command SHOULD include the <airsync:WindowSize> element, the recommended value for which is 100.<br>This step is repeated for each collection to be synchronized or all collections can be combined into one request. | The server response contains <airsync:Add>, <airsync:Change>, or <airsync:Delete> elements for items in the collection. If the response contains the <airsync:MoreAvailable> element, this step is repeated. |

The client can use the <airsync:WindowSize> element to break the server <airsync:Add> elements into multiple sets of items. The recommended window size is 100. For more details about the <airsync:WindowSize> element used by the **Sync** command, see section 2.2.2.19.1.2.1.1.6.

### 3.1.5.5 Receiving and Accepting Meeting Requests

This section describes how to retrieve items from the Inbox folder by using the **Sync** command, to respond to a meeting request item by using the **MeetingResponse** command, and to synchronize the Calendar folder by using the **Sync** command so that the new **Calendar object** is added to the client's calendar.

A meeting request is returned by the server in response to a synchronization of the Inbox folder. A meeting request is an e-mail message that has an embedded calendar item. The message contains an <email:MessageClass> element that has a value of IPM.Schedule.Meeting.Request, and its <airsync:ApplicationData> element contains an <email:MeetingRequest> element. When the client displays the meeting request message, the client SHOULD offer the options of accepting, declining, or tentatively accepting the meeting. If one of these actions is selected, the client sends a **MeetingResponse** command to the server.



Copyright © 2010 Microsoft Corporation.

If the response to the meeting is accepted or is tentatively accepted, the server will add or update the corresponding calendar item and return its server ID in the <meetingresponse:CalendarId> element of the response. If the response to the meeting is declined, the response will not contain a <meetingresponse:CalendarId> element because the server will delete the corresponding calendar item. If the client had created a tentative meeting **calendar** item, the client updates that item with the returned server ID (if accepted or tentative). The client MUST also change the busy status on the client calendar item from tentative to busy if the meeting request was accepted. Note that, if the client synchronizes the Calendar folder after responding to a meeting request, the calendar item in question will be in conflict if the client also sends the changed item change for it back to the server. This conflict is resolved according to the conflict resolution rules that are specified by the client in the **Sync** command request.

If the meeting request was accepted, the Calendar folder MUST be synchronized for the client to obtain the new calendar item. The new calendar item for the accepted meeting is added here and MUST be added to the client's calendar.

The following table lists the command sequence for receiving and accepting meeting requests. The asterisk (*) in the Order column means that a step can be repeated multiple times.

| Order | Client action | Server action |
|---|---|---|
| 1 | The client sends the **Sync** command for the Inbox collection with the value of the <airsync:SyncKey> element set to zero (0). | The server response contains the <airsync:SyncKey> for the collection, to be used in successive synchronizations. |
| 2* | The client sends a **Sync** command, specifying the <airsync:GetChanges> element and the <airsync:SyncKey> for the Inbox folder. The command SHOULD include the <airsync:WindowSize> element, the recommended value for which is 100. | The server response contains <airsync:Add> elements for items in the Inbox collection, including a meeting request item. If the response contains the <airsync:MoreAvailable> element, this step is repeated. |
| 3 | The user chooses to accept, decline, or tentatively accept a meeting request that is displayed in the client UI. | |
| 4 | The client sends a **MeetingResponse** command to the server, which specifies that the meeting was accepted, declined, or tentatively accepted, and provides the server IDs of the meeting request message and its parent folder. | The server sends a response that contains the **MeetingResponse** command request status along with the ID of the calendar item that corresponds to this meeting request if the meeting was not declined. |
| 5 | If a response was requested by the organizer, the client should use a **SendMail** command to send an appropriately formatted meeting response. | If the message was sent successfully, the server returns an empty response as specified in section 2.2.2.15.2. Otherwise, the server responds with a <Status> element that indicates the type of failure. |
| 6 | If the meeting was not declined, the client sends a **Sync** command for the calendar collection, specifying the <GetChanges> element. | The server responds with any changes to the Calendar folder caused by the last synchronization and the new calendar item for the accepted meeting. |

### 3.1.5.6 Handling Status Errors

The client MUST handle errors that occur during synchronization sessions. Errors fall into two categories: HTTP errors and ActiveSync protocol errors. HTTP errors are standard error codes, such



Copyright © 2010 Microsoft Corporation.

as 401 Logon failed, and they are returned from the server in response to an HTTP POST. ActiveSync protocol errors result from a problem on the server, in an attempt to perform the task requested by the command request message. ActiveSync protocol errors are indicated by codes that are returned in the <Status> element of a command response. For more details about the status codes, see section 2.2.3.

The client MUST implement error handling and a user interface (UI). Some errors are handled by a recovery procedure. Other errors require that an error message be displayed, along with a prompt for the user to respond. The client determines whether to run a recovery procedure or prompt for user input.

In addition to the ActiveSync protocol errors that the server sends, incomplete communication between server and client can result in the failure of a synchronization session. The server has an error recovery feature that enables a client to respond to errors by repeating the most recent synchronization session. The client MUST handle synchronization failures by retrying the synchronization. The server tracks synchronization requests to be able to respond appropriately in both of the following cases:

- The client failed in communicating a full request to the server for synchronization.
  - In this case, the client sends a request but the server does not receive the request. The server does not act on the request, and no server-side changes occur. Therefore, no response is sent to the client. The client MUST resend a synchronization request if there is no immediate server response and the <airsync:Wait> or <airsync:HeartbeatInterval> value was not sent in the **Sync** request, or if the <airsync:Wait> or <airsync:HeartbeatInterval> value was specified in the **Sync** request and the time has elapsed.
- The server failed in communicating a response to the client for updates.
  - In this case, the server response is not received by the client. The client knows it has not received a response if the <airsync:Wait> or <airsync:HeartbeatInterval> value was not sent in the **Sync** request and the server response is not received immediately, or if the <airsync:Wait> or <airsync:HeartbeatInterval> value was specified in the **Sync** request and that time has elapsed. The data on the server changed. The client MUST resend the request. The server recognizes the duplicate request. Because the server changes have already occurred, the server resends the response to the client to keep the server and client synchronized.

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.



Copyright © 2010 Microsoft Corporation.

# 4 Protocol Examples

## 4.1 Downloading the Current Server Security Policy by Using the Provision Command

For examples on downloading the current server security policy by using the **Provision** command, see [MS-ASPROV] section 4.

## 4.2 Discovering Account Settings by Using the Autodiscover Command

The **Autodiscover** command enables clients to discover core account configuration information by using the user's SMTP address as the primary input by means of the following process:

1. The end-user enters his or her e-mail address and domain credentials, for example, kim@contoso.com.
2. The client uses the domain information in the user's e-mail address, that is, contoso.com, and tries to locate the Autodiscover service by sending an **Autodiscover** command request to the following predefined URLs:

   https://<*SMTP-address-domain*>/autodiscover/autodiscover.xml

   https://autodiscover.<*SMTP-address-domain*>/autodiscover/autodiscover.xml

   In this example, these URLs map to https://contoso/autodiscover/autodiscover.xml and https://autodiscover.contoso/autodiscover/autodiscover.xml.
3. If the **domain name system (DNS)** contains a host record that maps one of these URLs to a server where the Autodiscover service is hosted, then the Autodiscover service responds with the settings that are required for the device to begin synchronizing. This response includes values for the Server type, the URL, and the Name element.
4. If redirection to another Autodiscover service is required, then the <Redirect> element is present and contains a URL to the **Autodiscover server** to query for the desired information.
5. The device then re-creates a partnership with the new server, and send an **Autodiscover** command request to that server.
6. If the response included the settings that are required for the device to begin synchronization, then the device applies the settings to initiate synchronization.
7. If the **Autodiscover** command request in step 3 fails, then the device performs a DNS SRV lookup for _autodiscover._tcp.<*smtp-address-domain*>.com, which in this example maps to _autodiscover._tcp.contoso.com. If the DNS lookup is successful, then "mail.<*smtp-address-domain*>.com" is returned, which maps to "mail.contoso.com". The device then applies the settings to initiatesynchronization. For more details about performing the DNS SRV lookup, see [MSFT-DNS-SRV].

The following sections show success and error response messages.

Account autodiscovery uses an e-mail address to look up information that is required to configure software. Given an e-mail name (such as EduardDell@Woodgrovebank.com), a list of possible Autodiscover servers is generated. The client contacts the name Autodiscover.domainname to provide the information. If that information is not found, the client tries to send the request to the domain name. If the information still is not retrieved, the client can use a manual configuration. For example, the client tries these servers:



Copyright © 2010 Microsoft Corporation.

- autodiscover.woodgrovebank.com
- woodgrovebank.com

Each server is sent an HTTP **POST** command. The post data is an XML request for a certain type of information. E-mail account configuration is the first use. The XML contains information that helps execute the request. For mail, the information includes the e-mail address, the protocols that the client software supports, the Web browser that is installed, the type of proxy that is being used, and the types of authentication that can be used.

The post is sent for servername/autodiscover/autodiscover.xml. The server name is defined according to the process described earlier in this topic.

### 4.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006">
   <Request>
      <EMailAddress>eduarddell@woodgrovebank.com</EMailAddress>
      <AcceptableResponseSchema>
      http://schemas.microsoft.com/exchange/autodiscover/mobilesync/
      responseschema/2006
      </AcceptableResponseSchema>
   </Request>
</Autodiscover>
```

### 4.2.2 Response - Case Error

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns:autodiscover="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responsesch
ema/2006">
    <autodiscover:Response>
        <autodiscover:Culture>en:us</autodiscover:Culture>
        <autodiscover:User>

<autodiscover:EMailAddress>eduarddell@woodgrovebank.com</autodiscover:EMailAddress>
       </autodiscover:User>
       <autodiscover:Action>
           <autodiscover:Error>
               <Status>1</Status>
               <Message>The directory service could not be reached</Message>
               <DebugData>MailUser</DebugData>
           </autodiscover:Error>
       </autodiscover:Action>
    </autodiscover:Response>
</Autodiscover>
```

### 4.2.3 Response - Case Redirect

In the following redirect example, assume that the **Autodiscover** command request was sent to autodiscover.woodgrovebank.com. The redirect node indicates to the client to try autodiscover.loandept.woodgrovebank.com.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns:autodiscover="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responsesch
ema/2006">
    <autodiscover:Response>
        <autodiscover:Culture>en:us</autodiscover:Culture>
        <autodiscover:User>
           <autodiscover:DisplayName>Eduard Dell</autodiscover:DisplayName>

<autodiscover:EMailAddress>eduarddell@woodgrovebank.com</autodiscover:EMailAddress>
        </autodiscover:User>
        <autodiscover:Action>
           <autodiscover:Redirect>eduarddell@loandept.woodgrovebank.com
</autodiscover:Redirect>
        </autodiscover:Action>
    </autodiscover:Response>
</Autodiscover>
```

### 4.2.4 Response - Case Server Settings

In the following success response, the Autodiscover service has provided server URL information for two services: MobileSync and CertEnroll. The client can use the MobileSync URL to configure the synchronization settings.. The client can also optionally use the CertEnroll information to obtain a client certificate for SSL negotiation.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns:autodiscover="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responsesch
ema/2006">
    <autodiscover:Response>
        <autodiscover:Culture>en:us</autodiscover:Culture>
        <autodiscover:User>
            <autodiscover:DisplayName>Eduard Dell</autodiscover:DisplayName>

<autodiscover:EMailAddress>eduarddell@woodgrovebank.com</autodiscover:EMailAddress>
        </autodiscover:User>
        <autodiscover:Action>
            <autodiscover:Settings>
                <autodiscover:Server>
                    <autodiscover:Type>MobileSync</autodiscover:Type>
                    <autodiscover:Url>
                        https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
                    </autodiscover:Url>
                    <autodiscover:Name>
                 https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
             </autodiscover:Name>
                </autodiscover:Server>
                <autodiscover:Server>
                    <autodiscover:Type>CertEnroll</autodiscover:Type>

<autodiscover:Url>https://cert.woodgrovebank.com/CertEnroll</autodiscover:Url>
                    <autodiscover:Name />
                   <autodiscover:ServerData>CertEnrollTemplate</autodiscover:ServerData>
                </autodiscover:Server>
            </autodiscover:Settings>
        </autodiscover:Action>
    </autodiscover:Response>
```



Copyright © 2010 Microsoft Corporation.

```
</Autodiscover>
```

### 4.2.5 Response - Case Framework Error

If the provider cannot be found, or if the <autodiscover:AcceptableResponseSchema> cannot be matched, then the following XML fragment is returned to the client.

The error code 600 means an invalid request was sent to the service, and 601 means that a provider could not be found to handle the <autodiscover:AcceptableResponseSchema> that was specified.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns:autodiscover="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responsesch
ema/2006">
   <autodiscover:Response>
      <autodiscover:Error Time="16:56:32.6164027" Id="1054084152">
          <autodiscover:ErrorCode>600</autodiscover:ErrorCode>
          <autodiscover:Message>Invalid Request</autodiscover:Message>
          <autodiscover:DebugData />
      </autodiscover:Error>
   </autodiscover:Response>
</Autodiscover>
```

### 4.2.6 Response – Case Framework Default

For unauthenticated requests, the server can create and serve a static page with contents to aid in troubleshooting errors, such as the following.<106>

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover>
  <Account>
    <AccountType>default e-mail</AccountType>
    <Action>settings</Action>
    <Image>http://www.abcd.com/def.jpg</Image>
    <ServiceHome>http://www.microsoft.com</ServiceHome>
    <RedirectUrl>...</RedirectUrl>

    <Protocol>
      <Type>POP</Type>
      <Server>popserverFQDN</Server>
      <Port>110</Port>
    </Protocol>

    <Protocol>
      <Type>SMTP</Type>
      <Server>smtpserverFQDN</Server>
      <Port>25</Port>
    </Protocol>

    <Protocol>
      <Type>IMAP</Type>
      <Server>imapserver1FQDN</Server>
    </Protocol>
```



Copyright © 2010 Microsoft Corporation.

```
    <Protocol>
      <Type>IMAP</Type>
      <Server>imapserver2FQDN</Server>
      <Port>143</Port>
    </Protocol>
  </Account>
</Autodiscover>
```

## 4.3 Setting Device Information by Using the Settings Command

The following example shows a device-information request and response.

### 4.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
    <DeviceInformation>
        <Set>
            <Model>...</Model>
            <IMEI>...</IMEI>
            <FriendlyName>...</FriendlyName>
            <OS>...</OS>
            <OSLanguage>...</OSLanguage>
            <PhoneNumber>...</PhoneNumber>
            <MobileOperator>...</MobileOperator>
            <UserAgent>...</UserAgent>
        </Set>
    </DeviceInformation>
</Settings>
```

### 4.3.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
    <Status>1</Status>
    <DeviceInformation>
        <Set>
            <Status>...</Status>
        </Set>
    </DeviceInformation>
</Settings>
```

## 4.4 Synchronizing Folders by Using the FolderSync Command

The following examples show the initial **FolderSync** command request sent from the client and the response sent from the server.

### 4.4.1 Request

The following example shows the initial **FolderSync** command request sent from the client to the server. The <folderhierarchy:SyncKey> element in the request is set to 0, because this is the first **FolderSync**. Any subsequent **FolderSync** requests should contain the <folderhierarchy:SyncKey> value returned in the previous **FolderSync** response.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <SyncKey>0</SyncKey>
</FolderSync>
```

### 4.4.2 Response

The following example shows the initial **FolderSync** command response sent from the server to the client. The <folderhierarchy:SyncKey> value has been incremented in the response, and this value should be used in the next **FolderSync**, **FolderCreate**, **FolderDelete**, or **FolderUpdate** request. Each folder in the collection is added to the client using an <folderhierarchy:Add> tag, which specifies the server ID of the folder, the parent ID of the folder, a display name, and the type of the folder, which indicates whether it is an e-mail, calendar, task, or other type of folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>1</SyncKey>
  <Changes>
    <Count>13</Count>
    <Add>
      <ServerId>1</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Calendar</DisplayName>
      <Type>8</Type>
    </Add>
    <Add>
      <ServerId>2</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Contacts</DisplayName>
      <Type>9</Type>
    </Add>
    <Add>
      <ServerId>3</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Deleted Items</DisplayName>
      <Type>4</Type>
    </Add>
    <Add>
      <ServerId>4</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Drafts</DisplayName>
      <Type>3</Type>
    </Add>
    <Add>
      <ServerId>5</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Inbox</DisplayName>
      <Type>2</Type>
    </Add>
    <Add>
      <ServerId>12</ServerId>
      <ParentId>5</ParentId>
      <DisplayName>NewFolder</DisplayName>
      <Type>12</Type>
```



Copyright © 2010 Microsoft Corporation.

```
      </Add>
      <Add>
        <ServerId>6</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Journal</DisplayName>
        <Type>11</Type>
      </Add>
      <Add>
        <ServerId>7</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Junk E-Mail</DisplayName>
        <Type>12</Type>
      </Add>
      <Add>
        <ServerId>8</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Notes</DisplayName>
        <Type>10</Type>
      </Add>
      <Add>
        <ServerId>9</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Outbox</DisplayName>
        <Type>6</Type>
      </Add>
      <Add>
        <ServerId>10</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Sent Items</DisplayName>
        <Type>5</Type>
      </Add>
      <Add>
        <ServerId>11</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Tasks</DisplayName>
        <Type>7</Type>
      </Add>
      <Add>
        <ServerId>RI</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>RecipientInfo</DisplayName>
        <Type>19</Type>
      </Add>
    </Changes>
  </FolderSync>
```

## 4.5 Synchronizing Data by Using the Sync Command

This section provides sample messages related to the **Sync** command.

### 4.5.1 Downloading Current Information from the Server

The following example shows a request that is sent to the server to synchronize an e-mail folder. The request asks that deleted items be moved to the Deleted Items folder. The request also asks for changes on the server to be specified in the response. The server response contains the new synchronization key and the items to be added, deleted, and changed on the client.



Copyright © 2010 Microsoft Corporation.

#### 4.5.1.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
    <Collections>
        <Collection>
            <Class>Email</Class>
            <SyncKey>6</SyncKey>
            <CollectionId>1</CollectionId>
            <DeletesAsMoves/>
            <GetChanges/>
            <Options> ... </Options>
        </Collection>
    </Collections>
</Sync>
```

#### 4.5.1.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
    <Collections>
        <Collection>
            <Class>Email</Class>
            <SyncKey>7</SyncKey>
            <CollectionId>1</CollectionId>
            <Status>1</Status>
            <Commands>
                <Add>...</Add>
                <Delete>...</Delete>
                <Change>...</Change>
                <Fetch>...</Fetch>
            </Commands>
        </Collection>
    </Collections>
</Sync>
```

### 4.5.2 Fetching an E-Mail by Using the ServerId

The following example shows a request that is sent to the server to fetch an item from the server by its <airsync:ServerId>.

#### 4.5.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
…
    <Commands>
        <Fetch >
            <ServerId>1:14</ServerId>
        </Fetch >
    </Commands>
…
</Sync>
```



Copyright © 2010 Microsoft Corporation.

#### 4.5.2.2 Response

A response from the server contains the server ID, status, and application data of the requested item.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
…
    <Responses>
        <Fetch>
            <ServerId>1:14</ServerId>
            <Status>1</Status>
            <ApplicationData>...</ApplicationData>
        </Fetch>
    </Responses>
…
</Sync>
```

### 4.5.3 Uploading New ApplicationData to the Server

This example shows a **Sync** command request that is sent to the server to add a contact. The response from the server shows that the synchronization was successful and that the new item from the client, identified by the <airsync:ClientId> element, was added to the collection on the server. The server also assigns a permanent ID for the newly added item in the <airsync:ServerId> element. After the client receives a successful response, the client uses this server ID for any future <airsync:Change> or <airsync:Delete> commands for this item.

#### 4.5.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:contacts="Contacts:" xmlns="AirSync:">
…
    <Commands>
        <Add>
            <ClientId>123</ClientId>
            <ApplicationData>
                <contacts:Email1Address>schai@fourthcoffee.com</contacts:Email1Address>
                <contacts:FirstName>Sean</contacts:FirstName>
                <contacts:MiddleName>W</contacts:MiddleName>
                <contacts:LastName>Chai</contacts:LastName>
                <contacts:Title>Sr Marketing Manager</contacts:Title>
            </ApplicationData>
        </Add>
    </Commands>
…
</Sync>
```

#### 4.5.3.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
…
    <Responses>
```



Copyright © 2010 Microsoft Corporation.

```
        <Add>
            <ClientId>123</ClientId>
            <ServerId>4:1</ServerId>
            <Status>1</Status>
        </Add>
    </Responses>
…
</Sync>
```

### 4.5.4 Updating ApplicationData on the Server

The following example shows a **Sync** command request from the client. The request modifies a contact, which is identified by the server ID, on the server. The response from the server shows that the change that is specified by the **Sync** request of the client succeeded and supplies the synchronization key and collection ID of the changed item.

#### 4.5.4.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:contacts="Contacts:" xmlns="AirSync:">
…
    <Commands>
        <Change>
            <ServerId>3:1</ServerId>
            <ApplicationData>
                <contacts:Email1Address>jsmith@fourthcoffee.com</contacts:Email1Address>
                <contacts:FirstName>Jeff</contacts:FirstName>
            </ApplicationData>
        </Change>
    </Commands>
…
</Sync>
```

#### 4.5.4.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
…
    <Collections>
        <Collection>
          <Class>Contacts</Class>
          <SyncKey>4</SyncKey>
          <CollectionId>1</CollectionId>
          <Status>1</Status>
        </Collection>
    </Collections>
…
</Sync>
```



Copyright © 2010 Microsoft Corporation.

### 4.5.5 Deleting an Item from the Server

The following examples show how to delete an e-mail message from the server by using the **Sync** command.

#### 4.5.5.1 Request

The following example shows a **Sync** command request sent from the client. The request deletes an e-mail message, which is identified by the <airsync:ServerId> element, from the server.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1537401391</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Commands>
        <Delete>
          <ServerId>5:1</ServerId>
        </Delete>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

#### 4.5.5.2 Response

The following example shows a **Sync** command response from the server. The response includes the incremented <airsync:SyncKey> value, a <airsync:Status> value of 1, indicating the success of the message deletion, and the <airsync:CollectionId> of the updated collection.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>609636509</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

### 4.5.6 Identifying Acceptance of Partial Collections

The following example shows a **Sync** request that includes the <Partial> element. The <Partial> element indicates that the current **Sync** request does not include all of the settings for all the collections to be synchronized, and that the server is to use the values from the previous **Sync** request.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
    <Collections>
        <Collection>
            <SyncKey>1723058747</SyncKey>
            <CollectionId>10</CollectionId>
        </Collection>
    </Collections>
    <Wait>8</Wait>
    <Partial/>
</Sync>
```

### 4.5.7 Identifying Acceptance of MIME Content

These examples use the <airsync:MIMESupport>, <airsync:MIMETruncation>, and <airsyncbase:Type> ([MS-ASAIRS] section 2.2.2.2.1) elements to identify acceptance of MIME content on the client.

#### 4.5.7.1 Sync Request With Support for MIME Content

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <MIMETruncation>1</MIMETruncation>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

#### 4.5.7.2 Sync Response with MIME Content

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email:" xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  …
  <Add>
     <ServerId>1:3</ServerId>
     <ApplicationData>
        <email:To>"James Smith" &lt;jsmith@contoso.com&gt;</email:To>
        <email:From>"Jyothi Pai" &lt;jpai@contoso.com&gt;</email:From>
        <email:Subject>RE: Presentation</email:Subject>
        <email:DateReceived>2004-11-12T00:45:06.000Z</email:DateReceived>
        <email:DisplayTo>James Smith</email:DisplayTo>
        <email:Importance>1</email:Importance>
        <email:Read>0</email:Read>
        <email:Importance>1</email:Importance>
        <email:Read>0</email:Read>
        <airsyncbase:Body>
```



Copyright © 2010 Microsoft Corporation.

```
                <airsyncbase:Type>4</airsyncbase:Type>
                <airsyncbase:EstimatedDataSize>1</airsyncbase:EstimatedDataSize>
                <airsyncbase:Truncated>5408</airsyncbase:Truncated>
                <airsyncbase:Data>
                   <!--Content removed -->
                </airsyncbase:Data>
          </airsyncbase:Body>
          <email:MessageClass>IPM.Note.SMIME.MultipartSigned</email:MessageClass>
          <email:InternetCPID>20127</email:InternetCPID>
          <email:Flag/>
          <email:ContentClass>urn:content-classes:message</email:ContentClass>
          <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
      </ApplicationData>
  </Add>
  …
</Sync>
```

### 4.5.7.3 Sync Request with BodyPreference and MIME Support

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:airsyncbase="AirSyncBase:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>17</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <airsyncbase:BodyPreference>
          <airsyncbase:Type>4</airsyncbase:Type>
          <airsyncbase:TruncationSize>512</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

### 4.5.7.4 Sync Response with MIME Support

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:email="Email:" xmlns:email2="Email2"
xmlns:airsyncbase="AirSyncBase:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>3</SyncKey>
      <CollectionId>17</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>17:11</ServerId>
          <ApplicationData>
            <email:To>"Mike Phipps" &lt;mike@contoso.com&gt;</email:To>
            <email:From>"Arlene Huff" &lt;arlene@contoso.com&gt;</email:From>
```



Copyright © 2010 Microsoft Corporation.

```
            <email:Subject>opaque s + e </email:Subject>
            <email:DateReceived>2007-02-01T06:42:46.015Z</email:DateReceived>
            <email:DisplayTo>Mike Phipps</email:DisplayTo>
            <email:ThreadTopic>opaque s + e</email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>1</email:Read>
            <airsyncbase:Attachments>
              <airsyncbase:Attachment>
                <airsyncbase:DisplayName>smime.p7m</airsyncbase:DisplayName>
                <airsyncbase:FileReference>17%3a11%3a0</airsyncbase:FileReference>
                <airsyncbase:Method>1</airsyncbase:Method>
                <airsyncbase:EstimatedDataSize>9340</airsyncbase:EstimatedDataSize>
              </airsyncbase:Attachment>
            </airsyncbase:Attachments>
            <airsyncbase:Body>
              <airsyncbase:Type>4</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>13814</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
              <airsyncbase:Data>Received: from contoso.com ([157.55.97.121])
              By contoso.com ([157.55.97.121]) with mapi;
              Wed, 31 Jan 2007 22:42:45 -0800
              From: Arlene Huff &lt;arlene@contoso.com&gt;
              To: Mike &lt;mike@contoso.com&gt;
              Content-Class: urn:content-classes:message
              Date: Wed, 31 Jan 2007 22:42:41 -0800
              Subject: opaque s + e
              Thread-Topic: opaque s + e
              Thread-Index: AcdFzCv5tyCXieBuTd2I5APpEvS+iQ==
              Message-ID:
              &lt;3AA64EB47EA90</airsyncbase:Data>
            </airsyncbase:Body>
            <email:MessageClass>IPM.Note.SMIME</email:MessageClass>
            <email:InternetCPID>20127</email:InternetCPID>
            <email:Flag/>
            <email:ContentClass>urn:content-classes:message</email:ContentClass>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
            <email2:ConversationId>…</email2:ConversationId>
            <email2:ConversationIndex>…</email2:ConversationIndex>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

### 4.5.8 Identifying That More Content is Ready for Download

The following example is a response message indicating that more content is available for download from the server. Because the content exceeded the <airsync:WindowSize> value, the <airsync:MoreAvailable> element is included in the response, indicating that additional content is available on the server for download.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
…
      <Collection>
            <Class>Email</Class>
```



Copyright © 2010 Microsoft Corporation.

```
            <SyncKey>2</SyncKey>
            <CollectionId>1</CollectionId>
            <Status>1</Status>
            <Commands>
                  ...
            </Commands>
            <MoreAvailable/>
      </Collection>
…
</Sync>
```

### 4.5.9 Synchronizing the Calendar Folder

The following example shows the initial synchronization of the Calendar folder, starting with a synchronization key of 0.

#### 4.5.9.1 Initial Request

The following example shows the initial **Sync** request command used to retrieve a nonzero <airsync:SyncKey> value for the specified collection. The nonzero <airsync:SyncKey> value is used by a second **Sync** command request to retrieve the contents of the collection. The <airsync:CollectionId> element identifies the Calendar folder as the collection being synchronized.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <WindowSize>100</WindowSize>
    </Collection>
  </Collections>
</Sync>
```

#### 4.5.9.2 Initial Response

The following example shows the initial **Sync** response command used to synchronize the Calendar folder. The <airsync:SyncKey> value returned in the response is nonzero and is used in the next request to retrieve the contents of the Calendar folder.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1024910360</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```



Copyright © 2010 Microsoft Corporation.

#### 4.5.9.3 Second Request

The following example shows the second **Sync** request command used to synchronize the Calendar folder. The nonzero <airsync:SyncKey> value from the previous **Sync** response is included in this request to retrieve the contents of the Calendar folder.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1024910360</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>100</WindowSize>
    </Collection>
  </Collections>
</Sync>
```

#### 4.5.9.4 Second Response

The following example shows the **Sync** response command used to synchronize the contents of the Calendar folder.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar:" xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1024910365</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>1:1</ServerId>
          <ApplicationData>

<calendar:TimeZone>aAEAACgARwBNAFQALQAwADYAOgAwADAAKQAgAEMAZQBuAHQAcgBhAGwAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAACgARwBNAFQALQAwADYAOgAwADAAKQAgAEMAZQBuAHQAcgBhAGw
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAAxP///w==</calendar:TimeZone>
            <calendar:DtStamp>20091020T161724Z</calendar:DtStamp>
            <calendar:StartTime>20091020T163000Z</calendar:StartTime>
            <calendar:Subject>Status Meeting</calendar:Subject>

<calendar:UID>040000008200E00074C5B7101A82E0080000000039F2FBCEA051CA0100000000000000000010000000
0D9221B7FF439514E87E0D2FE13295CE2</calendar:UID>
            <calendar:OrganizerName>Kim Akers</calendar:OrganizerName>
            <calendar:OrganizerEmail>kim@contoso.com</calendar:OrganizerEmail>
            <calendar:Attendees>
              <calendar:Attendee>
                <calendar:Email>Jeff@contoso.com</calendar:Email>
                <calendar:Name>Jeff Hay</calendar:Name>
                <calendar:AttendeeType>1</calendar:AttendeeType>
              </calendar:Attendee>
            </calendar:Attendees>
            <calendar:Location>My office</calendar:Location>
```



Copyright © 2010 Microsoft Corporation.

```
              <calendar:EndTime>20091020T173000Z</calendar:EndTime>
              <calendar:Recurrence>
                <calendar:Type>1</calendar:Type>
                <calendar:Interval>1</calendar:Interval>
                <calendar:DayOfWeek>4</calendar:DayOfWeek>
              </calendar:Recurrence>
              <airsyncbase:Body>
                <airsyncbase:Type>3</airsyncbase:Type>
                <airsyncbase:EstimatedDataSize>275</airsyncbase:EstimatedDataSize>
                <airsyncbase:Truncated>1</airsyncbase:Truncated>
              </airsyncbase:Body>
              <calendar:Sensitivity>0</calendar:Sensitivity>
              <calendar:BusyStatus>1</calendar:BusyStatus>
              <calendar:AllDayEvent>0</calendar:AllDayEvent>
              <calendar:Reminder>15</calendar:Reminder>
              <calendar:Exceptions>
                <calendar:Exception>
                  <calendar:DtStamp>20091020T162543Z</calendar:DtStamp>
                  <calendar:Location>Your office</calendar:Location>
                  <airsyncbase:Body>
                    <airsyncbase:Type>3</airsyncbase:Type>
                    <airsyncbase:EstimatedDataSize>273</airsyncbase:EstimatedDataSize>
                    <airsyncbase:Truncated>1</airsyncbase:Truncated>
                  </airsyncbase:Body>
                  <calendar:Categories />
                  <calendar:ExceptionStartTime>20091020T163000Z</calendar:ExceptionStartTime>
                  <calendar:BusyStatus>2</calendar:BusyStatus>

<calendar:AppointmentReplyTime>20091020T180257Z</calendar:AppointmentReplyTime>
                  <calendar:ResponseType>3</calendar:ResponseType>
                </calendar:Exception>
                <calendar:Exception>
                  <calendar:DtStamp>20091027T161136Z</calendar:DtStamp>
                  <calendar:StartTime>20091027T163000Z</calendar:StartTime>
                  <calendar:EndTime>20091027T173000Z</calendar:EndTime>
                  <calendar:Categories />
                  <calendar:ExceptionStartTime>20091027T163000Z</calendar:ExceptionStartTime>
                  <calendar:BusyStatus>2</calendar:BusyStatus>

<calendar:AppointmentReplyTime>20091027T161136Z</calendar:AppointmentReplyTime>
                  <calendar:ResponseType>3</calendar:ResponseType>
                </calendar:Exception>
              </calendar:Exceptions>
              <calendar:MeetingStatus>3</calendar:MeetingStatus>
              <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
              <calendar:ResponseRequested>1</calendar:ResponseRequested>
              <calendar:ResponseType>2</calendar:ResponseType>
            </ApplicationData>
          </Add>
          <Add>
            <ServerId>1:2</ServerId>
            <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGM
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAAxP///w==</calendar:TimeZone>
              <calendar:DtStamp>20091027T161226Z</calendar:DtStamp>
              <calendar:StartTime>20091029T163000Z</calendar:StartTime>
              <calendar:Subject>Planning meeting</calendar:Subject>
```



Copyright © 2010 Microsoft Corporation.

```
<calendar:UID>040000008200E00074C5B7101A82E008000000007D5639462057CA0100000000000000000010000000
065870766FB4D69479C217C16F9AD5E5B</calendar:UID>
            <calendar:OrganizerName>Kim Akers</calendar:OrganizerName>
            <calendar:OrganizerEmail>kim@contoso.com</calendar:OrganizerEmail>
            <calendar:Location />
            <calendar:EndTime>20091029T173000Z</calendar:EndTime>
            <airsyncbase:Body>
              <airsyncbase:Type>3</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>238</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <calendar:Sensitivity>0</calendar:Sensitivity>
            <calendar:BusyStatus>2</calendar:BusyStatus>
            <calendar:AllDayEvent>0</calendar:AllDayEvent>
            <calendar:Reminder>15</calendar:Reminder>
            <calendar:MeetingStatus>0</calendar:MeetingStatus>
            <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
            <calendar:ResponseRequested>1</calendar:ResponseRequested>
            <calendar:ResponseType>1</calendar:ResponseType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

### 4.5.10 Empty Sync Request and Response

This section demonstrates a scenario in which an empty **Sync** response and empty **Sync** request are exchanged between the client and server. For more details about empty **Sync** requests, see section 2.2.2.19.1.1. For more details about empty **Sync** responses, see section 2.2.2.19.2.2.

The scenario begins when a **Sync** request is issued by the client and indicates that there are no pending changes to report to the server. The **Sync** request is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>560109801</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
    </Collection>
  </Collections>
  <HeartbeatInterval>60</HeartbeatInterval>
  <WindowSize>512</WindowSize>
</Sync>
```

When the server receives this **Sync** request and determines that it contains no changes, it caches the request for future use. The server then responds to the **Sync** request with an empty **Sync** response when no changes or errors have occurred on the server. The empty **Sync** response is as follows:



Copyright © 2010 Microsoft Corporation.

```
HTTP/1.1 200 OK
Date: Fri, 10 Apr 2009 20:32:39 GMT
Content-Length: 0
```

When the client receives the empty **Sync** response, it can in turn send an empty **Sync** request if there are no pending changes. The empty **Sync** request is as follows:

```
POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=DeviceUser&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 2401271238
User-Agent: ASOM
Host: contoso.com
```

The exchange of the empty **Sync** requests and responses continues until a change is detected on either the client or server, at which time a **Sync** request or response with an XML payload is sent.

## 4.6 Sending E-Mail Messages by Using the SendMail Command

The following examples show how to send e-mail messages by using the **SendMail** command.

### 4.6.1 Request

The following example shows how a send an e-mail message by using the **SendMail** command request. The <composemail:ClientId> specifies the unique message ID of the message to be sent. The <composemail:SaveInSentItems> element indicates to the server that a copy of the message is placed in the Sent Items folder, and the <Mime> element contains the message contents.

```
<?xml version="1.0" encoding="utf-8"?>
<SendMail xmlns="ComposeMail:">
  <ClientId>633916348086136194</ClientId>
  <SaveInSentItems />
  <Mime>From: testuser1
To: testuser2
Cc:
Bcc:
Subject:
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the e-mail body content.</Mime>
</SendMail>
```

### 4.6.2 Response

The following example shows the successful **SendMail** command response.

```
HTTP/1.1 200 OK
Date: Wed, 24 Feb 2010 16:33:31 GMT
Content-Length: 0
```



Copyright © 2010 Microsoft Corporation.

## 4.7 Replying to E-Mail Messages by Using the SmartReply Command

The following examples show how to reply to e-mail messages by using the **SmartReply** command.

### 4.7.1 Request

The following example shows how to reply to an e-mail message by using the **SmartReply** command request. The <composemail:ClientId> specifies the unique message ID of the reply message to be sent. The <composemail:SaveInSentItems> element indicates to the server that a copy of the message is placed in the Sent Items folder. The <composemail:FolderId> element identifies the folder that contains the message that is being responded to, the <composemail:ItemId> element identifies the message being responded to, and the <composemail:Mime> element contains the message contents of the reply message.

```
<?xml version="1.0" encoding="utf-8"?>
<SmartReply xmlns="ComposeMail:">
  <ClientId>d7b99822-685a-4a40-8dfb-87a114926986</ClientId>
  <SaveInSentItems />
  <Source>
    <FolderId>5</FolderId>
    <ItemId>5:10</ItemId>
  </Source>
  <Mime>From: testuser1
To: "Test User 2" &lt;testuser2@contoso.com&gt;
Cc:
Bcc:
Subject: RE: Lunch
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the body of the smart reply message.</Mime>
</SmartReply>
```

### 4.7.2 Response

The following example shows the successful **SmartReply** command response. As specified in section 2.2.2.18.2, a successful **SmartReply** response contains no XML content.

```
HTTP/1.1 200 OK
Date: Wed, 22 Feb 2010 18:19:26 GMT
Content-Length: 0
```

## 4.8 Pinging the Server for Updates by Using the Ping Command

This section provides sample messages related to **Ping**.

### 4.8.1 Ping Command Request

The following is an example of a **Ping** command request in which the client is querying for new e-mail received in the folder that has an <ping:Id> value of 5.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
 <HeartbeatInterval>80</HeartbeatInterval>
  <Folders>
    <Folder>
      <Id>5</Id>
      <Class>Email</Class>
    </Folder>
  </Folders>
</Ping>
```

### 4.8.2 Ping Command Response

#### 4.8.2.1 Typical Response

The following example shows a typical response to a **Ping** command request, when the heartbeat interval that was specified by the client has expired and there were no changes in any of the specified folders.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
  <Status>1</Status>
</Ping>
```

#### 4.8.2.2 Response – Changes Found

The following response message shows that changes have occurred in two folders that were being monitored. The client then synchronizes the specified folders. Do not reissue the next **Ping** command until the folders have been synchronized.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
  <Status>2</Status>
  <Folders>
    <Folder>1234</Folder>
    <Folder>5678</Folder>
  </Folders>
</Ping>
```

#### 4.8.2.3 Response – HeartbeatInterval Error

The following example shows a response to a **Ping** command request that specified a heartbeat interval outside the acceptable range. The returned heartbeat interval is either the minimum or maximum allowed value. The client compares the requested interval with the returned interval and determine whether the requested heartbeat interval was either too great or too small.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
  <Status>5</Status>
  <HeartbeatInterval>60</HeartbeatInterval>
</Ping>
```



Copyright © 2010 Microsoft Corporation.

#### 4.8.2.4 Response – Folder Error

The following example shows a response to a **Ping** command request where the number of folders that was specified was greater than the maximum number of folders that are allowed to be monitored. The maximum number of folders that are allowed to be monitored is returned in the <ping:MaxFolders> element.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
  <Status>6</Status>
  <MaxFolders>200</MaxFolders>
</Ping>
```

## 4.9 Retrieving Item Estimates by Using the GetItemEstimate Command

The following examples show how to retrieve the number of items in a folder that require synchronization by using the **GetItemEstimate** command.

### 4.9.1 Request

The following example shows how the **GetItemEstimate** command request is used to retrieve the number of items in a folder that require synchronization. The <airsync:SyncKey> element in the request is set to the value returned by the last **Sync** command response. The <getitemestimate:CollectionId> identifies the folder to query, and the <airsync:Options> elements specify any time filters or specific content class items to retrieve.

```
<?xml version="1.0" encoding="utf-8"?>
<GetItemEstimate xmlns="GetItemEstimate:" xmlns:airsync="AirSync:">
  <Collections>
    <Collection>
      <airsync:SyncKey>785971245</airsync:SyncKey>
          <CollectionId>5</CollectionId>
          <airsync:Options>
              <airsync:FilterType>0</airsync:FilterType>
              <airsync:Class>SMS</airsync:Class>
          </airsync:Options>
    </Collection>
  </Collections>
</GetItemEstimate>
```

### 4.9.2 Response

The following example shows the **GetItemEstimate** command response. A <getitemestimate:Status> value of 1 is returned to indicate that the **GetItemEstimate** request was completed successfully. The <getitemestimate:CollectionId> element identifies the folder that was queried, and the <getitemestimate:Estimate> element specifies the number of items that require synchronization in the folder.

```
<?xml version="1.0" encoding="utf-8"?>
<GetItemEstimate xmlns="GetItemEstimate:">
  <Response>
```



Copyright © 2010 Microsoft Corporation.

```
      <Status>1</Status>
      <Collection>
        <CollectionId>5</CollectionId>
        <Estimate>1</Estimate>
      </Collection>
    </Response>
  </GetItemEstimate>
```

## 4.10 Fetching E-Mail and Attachments by Using the ItemOperations Command

The **ItemOperations** command enables the client to retrieve PIM items and attachments (in addition to document library items and search results) outside the **Sync** command context.

These examples focus on retrieval of items and attachments, following a simple request and response model. The following figure shows the request and response model used in fetching e-mail and attachments.

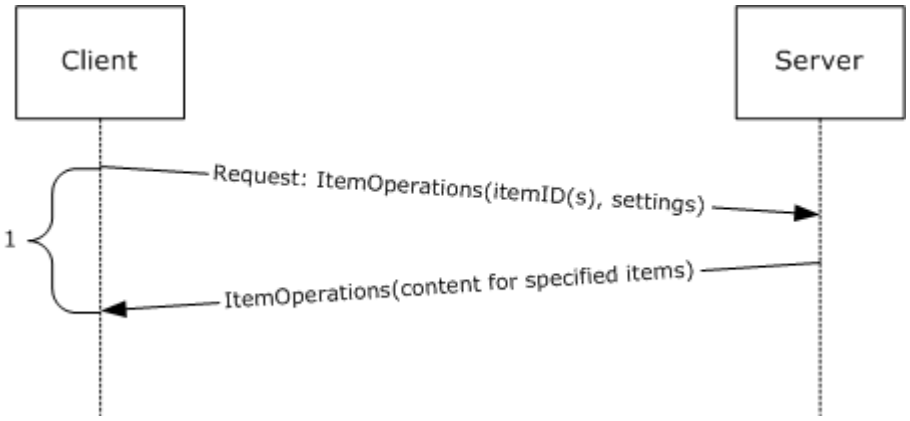


**Figure 7: Fetching E-mail**

### 4.10.1 Fetching an E-Mail Item

The following example shows the client retrieving an e-mail message by using the **ItemOperations** command.

#### 4.10.1.1 Request

```
<?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>
```



Copyright © 2010 Microsoft Corporation.

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

#### 4.10.1.2 Response

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync:" xmlns:email="Email:" xmlns:email2="Email2:"
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>
        <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
        <email2:ConversationId>…</email2:ConversationId>
        <email2:ConversationIndex>…</email2:ConversationIndex>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

### 4.10.2 Fetching a MIME E-Mail Item

The following example shows the client retrieving a MIME e-mail message by using the <airsync:MIMESupport> option.

#### 4.10.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
```



Copyright © 2010 Microsoft Corporation.

```
<ItemOperations xmlns="ItemOperations:" xmlns:airsync="AirSync:"
xmlns:airsyncbase="AirSyncBase:">
   <Fetch>
      <Store>Mailbox</Store>
      <airsync:CollectionId>17</airsync:CollectionId>
      <airsync:ServerId>17:11</airsync:ServerId>
      <Options>
         <airsync:MIMESupport>1</airsync:MIMESupport>
         <airsyncbase:BodyPreference>
            <airsyncbase:Type>4</airsyncbase:Type>
         </airsyncbase:BodyPreference>
      </Options>
   </Fetch>
</ItemOperations>
```

#### 4.10.2.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations:" xmlns:airsync="AirSync:" xmlns:email="Email:"
xmlns:email2="Email2:" xmlns:airsyncbase="AirSyncBase:">
   <Status>1</Status>
   <Response>
      <Fetch>
         <Status>1</Status>
         <airsync:CollectionId>17</airsync:CollectionId>
         <airsync:ServerId>17:11</airsync:ServerId>
         <airsync:Class>Email</airsync:Class>
         <Properties>
            <email:To>"Mike Phipps" &lt;mike@contoso.com&gt;</email:To>
            <email:From>"Arlene Huff" &lt;arlene@contoso.com&gt;</email:From>
            <email:Subject>opaque s + e</email:Subject>
            <email:DateReceived>2007-02-01T06:42:46.015Z</email:DateReceived>
            <email:DisplayTo>Mike Phipps</email:DisplayTo>
            <email:ThreadTopic>opaque s + e</email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>1</email:Read>
            <airsyncbase:Attachments>
               <airsyncbase:Attachment>
                  <airsyncbase:DisplayName>smime.p7m</airsyncbase:DisplayName>

<airsyncbase:FileReference>RgAAAAA4u8%2fWvU8lQ7GtLlC7V9V3BwCdyWYIRkOHRp2ozB%2f0DXQsAHgM%2bwAF
AAA6pk60fqkEQbWH4Wm%2bnjh7AHgNBA%2bgAAAJ%3a0</airsyncbase:FileReference>
                  <airsyncbase:Method>1</airsyncbase:Method>
                  <airsyncbase:EstimatedDataSize>9340</airsyncbase:EstimatedDataSize>
               </airsyncbase:Attachment>
            </airsyncbase:Attachments>
            <airsyncbase:Body>
               <airsyncbase:Type>4</airsyncbase:Type>
               <airsyncbase:EstimatedDataSize>13813</airsyncbase:EstimatedDataSize>
               <airsyncbase:Data>Received: from contoso.com ([157.55.97.121])
 by contoso.com ([157.55.97.121]) with mapi;
 Wed, 31 Jan 2007 22:42:45 -0800
From: Arlene Huff &lt;arlene@contoso.com&gt;
To: Mike Phipps &lt;mike@contoso.com&gt;
Content-Class: urn:content-classes:message
Date: Wed, 31 Jan 2007 22:42:41 -0800
Subject: opaque s + e
Thread-Topic: opaque s + e
```



Copyright © 2010 Microsoft Corporation.

```
Thread-Index: AcdFzCv5tyCXieBuTd2I5APpEvS+iQ==
Message-ID:
 &lt;3AA64EB47EA90441B587E169BE9E387B780D00C326@contoso.com&gt;
Content-Language: en-US
X-MS-Exchange-Organization-AuthAs: Internal
X-MS-Exchange-Organization-AuthMechanism: 04
X-MS-Exchange-Organization-AuthSource:
 contoso.com
X-MS-Has-Attach: yes
X-MS-Exchange-Organization-SCL: -1
X-MS-TNEF-Correlator:
acceptlanguage: en-US
Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data;
name="smime.p7m"
Content-Disposition: attachment; filename="smime.p7m"
Content-Transfer-Encoding: base64
MIME-Version: 1.0

MIAGCSqGSIb3DQEHA6CAMIACAQAxggJEMIIBHgIBADCBhjB4MRMwEQYKCZImiZPyLGQBGRYDY29t
MRkwFwYKCZImiZPyLGQBGRYJbWljcm9zb2Z0MRYwFAYKCZImiZPyLGQBGRYGZXh0ZXN0MR0wGwYK
CZImiZPyLGQBGRYNamluZ2h1YWMwMURPTTEPMA0GA1UEAxMGVGVzdENBAgonJIo2AAAAAAAHMA0G
(Large section of sample data removed)
              </airsyncbase:Data>
            </airsyncbase:Body>
            <email:MessageClass>IPM.Note.SMIME</email:MessageClass>
            <email:InternetCPID>20127</email:InternetCPID>
            <email:Flag/>
            <email:ContentClass>urn:content-classes:message</email:ContentClass>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
            <email2:ConversationId>…</email2:ConversationId>
            <email2:ConversationIndex>…</email2:ConversationIndex>
         </Properties>
      </Fetch>
   </Response>
</ItemOperations>
```

### 4.10.3 Fetching an E-Mail Item with a LongId

The following example shows the client retrieving an e-mail message by using <search:LongId>. First, use the **Search** command to get the <search:LongId> of the message, and then use the <itemoperations:Fetch> element with the <search:LongId> element to retrieve the message.

#### 4.10.3.1 Search Request

The client sends the **Search** command request message, and it is searching for e-mails containing the text "Sales Totals".

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:airsync="AirSync:"
xmlns:email="Email:">
  <Store>
    <Name>Mailbox</Name>
    <Query>
      <And>
        <airsync:Class>Email</airsync:Class>
        <airsync:CollectionId>7</airsync:CollectionId>
```



Copyright © 2010 Microsoft Corporation.

```
        <FreeText>Sales Totals</FreeText>
      </And>
    </Query>
    <Options>
      <RebuildResults />
      <Range>0-4</Range>
    </Options>
  </Store>
</Search>
```

### 4.10.3.2 Search Response

The server sends the **Search** command response message which includes e-mail data for e-mail that contains the string "Sales Totals". Included with the results is the <search:LongId> element.

```
<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync:"
xmlns:email="Email:" xmlns:email2="Email2:" xmlns:airsyncbase="AirSyncBase:"
xmlns="Search:">
  <Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <airsync:Class>Email</airsync:Class>
        <LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7zU
        NyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</LongId>
        <airsync:CollectionId>7</airsync:CollectionId>
        <Properties>
          <email:To>"deviceuser" &lt;someone1@example.com&gt;
          </email:To>
          <email:From>"deviceuser2" &lt;someone2@example.com&gt;
          </email:From>
          <email:Subject>Sales Totals for April</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>1</email:Read>
          <airsyncbase:Body>
            <airsyncbase:Type>1</airsyncbase:Type>
            <airsyncbase:EstimatedDataSize>6
            </airsyncbase:EstimatedDataSize>
            <airsyncbase:Truncated>1</airsyncbase:Truncated>
          </airsyncbase:Body>
          <email:MessageClass>IPM.Note</email:MessageClass>
          <email:InternetCPID>28591</email:InternetCPID>
          <email:Flag />
          <email:ContentClass>urn:content-classes:message
          </email:ContentClass>
          <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          <email2:ConversationId>…</email2:ConversationId>
          <email2:ConversationIndex>…</email2:ConversationIndex>
        </Properties>
      </Result>
      <Range>0-0</Range>
```



Copyright © 2010 Microsoft Corporation.

```
      <Total>1</Total>
    </Store>
  </Response>
</Search>
```

### 4.10.3.3 Fetch Request

The **ItemOperations** command request, with the <itemoperations:Fetch> element is now sent by the client, and includes the <search:LongId> element retrieved by the **Search** command.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync:"
xmlns:airsyncbase="AirSyncBase:" xmlns:search="Search:"
xmlns="ItemOperations:">
  <Fetch>
    <Store>Mailbox</Store>
    <search:LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpc
    EQ7zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HA
    CAAAJ</search:LongId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
      </airsyncbase:BodyPreference>
    </Options>
  </Fetch>
</ItemOperations>
```

### 4.10.3.4 Fetch Response

The server sends the **ItemOperations** command response, with the <itemoperations:Fetch> element, which contains the complete e-mail for the specified message.

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync:" xmlns:email="Email:" xmlns:email2="Email2:"
xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <search:LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7zU
        NyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</search:LongId>
      <airsync:Class>Email</airsync:Class>
      <Properties>
        <email:To>"deviceuser" &lt;someone1@example.com&gt;</email:To>
        <email:From>"deviceuser2" &lt;someone2@example.com&gt;
        </email:From>
        <email:Subject>Sales Totals for April</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>1</email:Read>
        <airsyncbase:Body>
```



Copyright © 2010 Microsoft Corporation.

```
            <airsyncbase:Type>1</airsyncbase:Type>
            <airsyncbase:EstimatedDataSize>20
            </airsyncbase:EstimatedDataSize>
            <airsyncbase:Data>Income generated by the sales department
            in April can be attributed to the following...
            </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>
          <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          <email2:ConversationId>…</email2:ConversationId>
          <email2:ConversationIndex>…</email2:ConversationIndex>
        </Properties>
      </Fetch>
    </Response>
  </ItemOperations>
```

### 4.10.4 Fetching an Attachment

In the following example, the **Sync** command is used to synchronize a new message with an attachment to the client. Then, the **ItemOperations** command is used to retrieve the attachment.

In the XML schema code, HTML strings are escaped by using < and >. However, as these values are passed over the wire, they are passed in their original HTML format, as < and >.

#### 4.10.4.1 Sync Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves />
      <GetChanges />
    </Collection>
  </Collections>
</Sync>
```

#### 4.10.4.2 Sync Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>2</SyncKey>
      <CollectionId>7</CollectionId>
      <Status>1</Status>
      <Commands>
```



Copyright © 2010 Microsoft Corporation.

```
          <Add>
            <ServerId>7:1</ServerId>
            <ApplicationData>
              <email:To>"deviceuser" &lt;someone@example.com&gt;
              </email:To>
              <email:From>"deviceuser2" &lt;someone2@example.com&gt;
              </email:From>
              <email:Subject>Email with Attachment</email:Subject>
              <email:DateReceived>2007-05-08T17:57:22.890Z
              </email:DateReceived>
              <email:DisplayTo>deviceuser</email:DisplayTo>
              <email:ThreadTopic>Email with Attachment
              </email:ThreadTopic>
              <email:Importance>1</email:Importance>
              <email:Read>0</email:Read>
              <airsyncbase:Attachments>
                <airsyncbase:Attachment>
                  <airsyncbase:DisplayName>ActiveSyncClient_
                  AcceptingMeetingRequest.JPG</airsyncbase:DisplayName>
                  <airsyncbase:FileReference>7%3a1%3a0
                  </airsyncbase:FileReference>
                  <airsyncbase:Method>1</airsyncbase:Method>
                  <airsyncbase:EstimatedDataSize>18790
                  </airsyncbase:EstimatedDataSize>
                </airsyncbase:Attachment>
              </airsyncbase:Attachments>
              <airsyncbase:Body>
                <airsyncbase:Type>2</airsyncbase:Type>
                <airsyncbase:EstimatedDataSize>58
                </airsyncbase:EstimatedDataSize>
                <airsyncbase:Truncated>1</airsyncbase:Truncated>
                <airsyncbase:Data>&lt;html&gt;</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>
              <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
            </ApplicationData>
          </Add>
        </Commands>
      </Collection>
    </Collections>
  </Sync>
```

### 4.10.4.3 ItemOperation Request

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsyncbase="AirSyncBase:"
xmlns="ItemOperations:">
  <Fetch>
    <Store>Mailbox</Store>
       <airsyncbase:FileReference>7%3a1%3a0</airsyncbase:FileReference>
  </Fetch>
</ItemOperations>
```



Copyright © 2010 Microsoft Corporation.

#### 4.10.4.4 ItemOperation Response

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsyncbase="AirSyncBase:"
xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <airsyncbase:FileReference>7%3a1%3a0</airsyncbase:FileReference>
      <Properties>
        <airsyncbase:ContentType>text/plain
        </airsyncbase:ContentType>
        <Data>U291cmNlIERlcG90IFN5c3RlbSBSZXF1aXJlbWVudHMNClRv
        IHJ1biBTb3VyY2UgRGVwb3QsIHlvdXIgY29tcHV0ZXIgbXVzdCBtZ
        WV0IGNlcnRhaW4gaGFyZHdhcmUgYW5kIHNvZnR3YXJlIHJlcXVpc
        mVtZW50cy4gVG8gcnVuIHRoZSBTb3VyY2UgRGVwb3QgY2xpZW50L
        CBjaGVjayB0aGUgZm9sbG93aW5nIGxpc3Qgb2YgbWluaW11bSBw
        bGF0Zm9ybS1zcGVjaWZpYyByZXF1aXJlbWVudHMgZm9yIFNvdXJj
        ZSBEZXBvdC4gDQoNClNvZnR3YXJlIFJlcXVpcmVtZW50cw0KV2luZG9
        3cyBYUCBhbmQgV2luZG93cyAyMDAwIJYgSW50ZWwgb3IgQVhQ
        NjQNCk5UNCAtIEludGVsIA0KV2luZG93cyA5eCCWIEludGVsDQ
        pIYXJkd2FyZSBSZXF1aXJlbWVudHMNCkludGVsIENsaWVudA0K
        DQp4ODYgcHJvY2Vzc29yDQpQZW50aXVtIG9yIGJldHRlciANCkV
        ub3VnaCBoYXJkIGRpc2sgc3BhY2UgdG8gaG9sZCB0aGUgZmlsZX
        MNCkFYUDY0IENsaWVudA0KDQpBbHBoYSBDbGllbnQgc3lzdGVtI
        HJ1bm5pbmcgYW4gQVhQNjQgZW11bGF0b3IuIA0KSG93IHRvDQoN
        Ckluc3RhbGwgU291cmNlIERlcG90DQoNCg0KLS0tLS0tLS0tLS0t
        LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0
        tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0NCg0KU2VuZCB
        1cyBmZWVkYmFjayBhbmQgc3VnZ2VzdGlvbnMgb24gdGhpcyBIZW
        xwIHRvcGljIG9yIHRoaXMgcHJvZHVjdC4NCg0KQ29udGFjdCBzdX
        Bwb3J0IGZvciBhc3Npc3RhbmNlLg0KDQoNCmMuIDIwMDQuIE1pY3
        Jvc29mdCBDb25maWRlbnRpYWwgDQpMYXN0IHVwZGF0ZWQgMy8xO
        S8yMDA0IA==</Data>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

## 4.11 Searching for an Item in the Mailbox by Using the Search Command

This section provides sample messages used to perform keyword searches and forward search results for items in the mailbox.

### 4.11.1 Keyword Search

In the following example, the client is searching the Inbox in the mailbox by using the keyword "Presentation". The client has asked for the first 5 results and specified that it wants text bodies returned for the results. Note that the content of the <search:FreeText> element is not case-sensitive.

#### 4.11.1.1 Request



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:airsync="AirSync:">
<Store>
  <Name>Mailbox</Name>
    <Query>
      <And>
        <airsync:CollectionId>7</airsync:CollectionId>
        <FreeText>Presentation</FreeText>
      </And>
    </Query>
    <Options>
      <RebuildResults />
      <Range>0-4</Range>
      <DeepTraversal/>
    </Options>
  </Store>
</Search>
```

#### 4.11.1.2 Response

```
<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync:"
xmlns:email="Email:" xmlns:airsyncbase="AirSyncBase:"
xmlns="Search:">
<Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <airsync:Class>Email</airsync:Class>
        <LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7
        zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</LongId>
        <airsync:CollectionId>7</airsync:CollectionId>
        <Properties>
          <email:To>"deviceuser" &lt;someone1@example.com&gt;
          </email:To>
          <email:From>"deviceuser2"&lt;someone2@example.com&gt;
          </email:From>
          <email:Subject>Presentation</email:Subject>
          <email:DateReceived>2007-05-08T17:41:58.000Z
          </email:DateReceived>
          <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
          <A2:ThreadTopic>Presentation</A2:ThreadTopic>
          <A2:Importance>1</A2:Importance>
          <email:Read>1</email:Read>
          <airsyncbase:Body>
            <airsyncbase:Type>1</airsyncbase:Type>
            <airsyncbase:EstimatedDataSize>6
            </airsyncbase:EstimatedDataSize>
            <airsyncbase:Truncated>1</airsyncbase:Truncated>
          </airsyncbase:Body>
          <email:MessageClass>IPM.Note</email:MessageClass>
          <email:InternetCPID>28591</email:InternetCPID>
          <email:Flag />
          <email:ContentClass>urn:content-classes:message
          </email:ContentClass>
          <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
```



Copyright © 2010 Microsoft Corporation.

```
          <email2:ConversationId>Í÷Žo□7D´½□t (îD</email2:ConversationId>
          <email2:ConversationIndex>Ë&#x1E;Õ~%</email2:ConversationIndex>
          <email2:LastVerbExecuted>3</email2:LastVerbExecuted>
          <email2:LastVerbExecutionTime>2010-07-
12T19:15:25.446Z</email2:LastVerbExecutionTime>
          <email:Categories/>
        </Properties>
      </Result>
      <Range>0-0</Range>
      <Total>1</Total>
    </Store>
  </Response>
</Search>
```

### 4.11.2 Forward a Search Result

The client can then take the <composemail:LongId> for any given search result and forward the item.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<SmartForward xmlns="ComposeMail:">
  <ClientId>634145285982398784</ClientId>
  <Source>

<LongId>RgAAAAA%2fygDboRrKQ6odHngUY8KtBwCwZCmLZplDQJqLAuQKTJwjAAAAjGzeAACwZCmLZplDQJqLAuQKTJw
jAAAAjmvbAAAJ</LongId>
  </Source>
  <Mime>From: someone2
To: someone3@example.com
Cc:
Bcc:
Subject: Presentation
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the content of the forwarded message.</Mime>
</SmartForward>
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 12 Jul 2010 19:15:26 GMT
Content-Length: 0
```

### 4.11.3 Keyword Search with MIMESupport

In the following example, the client is searching the Inbox in the mailbox by using the keyword "text". The client has specified that it wants MIME data sent for all results.

Request



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:airsyncbase="AirSyncBase:" xmlns:airsync="AirSync:">
   <Store>
      <Name>Mailbox</Name>
      <Query>
         <And>
            <airsync:Class>Email</airsync:Class>
            <FreeText>text</FreeText>
         </And>
      </Query>
      <Options>
         <RebuildResults/>
         <DeepTraversal/>
         <Range>0-999</Range>
         <airsyncbase:BodyPreference>
            <airsyncbase:Type>4</airsyncbase:Type>
            <airsyncbase:TruncationSize>1024</airsyncbase:TruncationSize>
            <airsyncbase:Preview>120</airsyncbase:Preview>
         </airsyncbase:BodyPreference>
         <airsync:MIMESupport>2</airsync:MIMESupport>
      </Options>
   </Store>
</Search>
```

## Response

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:airsync="AirSync:" xmlns:email="Email:" xmlns:airsyncbase="AirSyncBase:"
xmlns="Search:">
   <Status>1</Status>
   <Response>
      <Store>
         <Status>1</Status>
         <Result>
            <airsync:Class>Email</airsync:Class>

<LongId>RgAAAAAaty%2f%2b4QxHTJOZnIR0P9qkBwA6pk60fqkEQbWH4Wm%2bnjh7AJKAUQo6AAA6pk60fqkEQbWH4Wm
%2bnjh7AJKAURrEAAAJ</LongId>
            <airsync:CollectionId>6</airsync:CollectionId>
            <Properties>
               <email:To>"NSyncUser1" &lt;NSyncUser1@contoso.com></email:To>
               <email:From>"NSyncUser1" &lt;NSyncUser1@contoso.com></email:From>
               <email:Subject>Subject</email:Subject>
               <email:DateReceived>2007-04-02T19:20:32.000Z</email:DateReceived>
               <email:DisplayTo>NSyncUser1</email:DisplayTo>
               <email:Read>1</email:Read>
               <airsyncbase:Body>
                  <airsyncbase:Type>4</airsyncbase:Type>
                  <airsyncbase:Preview>The beginning of this message</airsyncbase:Preview>
                  <airsyncbase:EstimatedDataSize>2288</airsyncbase:EstimatedDataSize>
                  <airsyncbase:Truncated>1</airsyncbase:Truncated>
                  <airsyncbase:Data>Received: from 157.55.97.120 ([157.55.97.120]) by
contoso.com ([157.55.97.121]) with Microsoft Exchange Server HTTP-DAV ; Mon, 2 Apr 2007
19:20:32 +0000 From: NSyncUser1 &lt;NSyncUser1@contoso.com> To: NSyncUser1
&lt;NSyncUser1@contoso.com> Content-Class: urn:content-classes:message Date: Mon, 27 Apr 1998
13:05:29 -0700 Subject: Subject Thread-Topic: Topic Message-ID:
&lt;3AA64EB47EA90441B587E169BE9E387B9280511AC4@contoso.com> Accept-Language: en-US X-MS-Has-
Attach: X-MS-TNEF-Correlator: Content-Type: text/plain; charset="iso-8859-1" Content-
Transfer-Encoding: quoted-printable MIME-Version: 1.0
```



Copyright © 2010 Microsoft Corporation.

```
Body12345678901234567890123456789012345678901234567890123456789012345678901=
23456789012345678901234567890123456789012345678901234567890123456789012345=
78901234567890123456789012345678901234567890123456789012345678901234567890=
23456789012345678901234567890123456789012345678901234567890123456789</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>
               <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
            </Properties>
         </Result>
         <Range>0-0</Range>
         <Total>1</Total>
      </Store>
   </Response>
</Search>
```

## 4.12 Searching the Global Address List by Using the Search Command

The following examples show how to search the GAL using the **Search** command.

### 4.12.1 Request

The following example shows a **Search** command request sent from the client to the server. The <search:Name> element identifies the GAL as the store to search, and the <search:Query> element identifies the search string as "Sandeep". Within the <search:Options> element, the <search:Range> element indicates that a maximum of two results can be returned in the response, the <search:RebuildResults> element specifies that the server will rebuild the search folder, and the <search:DeepTraversal> element specifies that all subfolders are searched.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:">
  <Store>
    <Name>GAL</Name>
    <Query>Sandeep</Query>
    <Options>
        <Range>0-1</Range>
        <RebuildResults/>
        <DeepTraversal/>
    </Options>
  </Store>
</Search>
```

### 4.12.2 Response

The following example shows the **Search** command response sent from the server to the client. A <search:Status> value of 1 is returned to indicate that the search was successful. The <search:Result> elements contain the GAL entries for the first two results that met the search criteria.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:gal="Gal:" xmlns="Search:">
```



Copyright © 2010 Microsoft Corporation.

```
    <Status>1</Status>
    <Response>
      <Store>
        <Status>1</Status>
        <Result>
          <Properties>
            <gal:DisplayName>Sandeep Kaliyath</gal:DisplayName>
            <gal:Phone>+1 (301) 5550156 X8376</gal:Phone>
            <gal:Office>Bldg36/6163</gal:Office>
            <gal:Title>SDE</gal:Title>
            <gal:Company>Contoso</gal:Company>
            <gal:Alias>sandeepk</gal:Alias>
            <gal:FirstName>Sandeep</gal:FirstName>
            <gal:LastName>Kaliyath</gal:LastName>
            <gal:EmailAddress>sandeepk@contoso.com</gal:EmailAddress>
          </Properties>
        </Result>
        <Result>
          <Properties>
            <gal:DisplayName>Sandeep Katyal</gal:DisplayName>
            <gal:Phone>+1 (953) 5550195 </gal:Phone>
            <gal:Office>Bldg2/2710</gal:Office>
            <gal:Title>SDET</gal:Title>
            <gal:Company>Contoso</gal:Company>
            <gal:Alias>sandeepka</gal:Alias>
            <gal:FirstName>Sandeep</gal:FirstName>
            <gal:LastName>Katyal</gal:LastName>
            <gal:MobilePhone>+1 (953) 5550167</gal:MobilePhone>
            <gal:EmailAddress>sandeepka@contoso.com</gal:EmailAddress>
          </Properties>
        </Result>
        <Range>0-1</Range>
        <Total>11</Total>
      </Store>
    </Response>
  </Search>
```

## 4.13 Working with Folders

This section contains examples that demonstrate how to create, delete, and update folders as well as how to empty folder contents.

### 4.13.1 Creating Folders by Using the FolderCreate Command

The following examples show how to create a folder using the **FolderCreate** command.

#### 4.13.1.1 Request

The following example shows a **FolderCreate** command request sent from the client to the server. The <folderhierarchy:SyncKey> element in the request is set to 1, which is the <folderhierarchy:SyncKey> value returned in the last **FolderSync** response (section 4.4.2). The <folderhierarchy:ParentId> element specifies that this is a child of the Inbox folder, the <folderhierarchy:Type> element specifies that this is an e-mail folder, and the <folderhierarchy:DisplayName> element identifies the friendly name of the folder as "NewFolder".



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderCreate xmlns="FolderHierarchy:">
  <SyncKey>1</SyncKey>
  <ParentId>5</ParentId>
  <DisplayName>NewFolder</DisplayName>
  <Type>12</Type>
</FolderCreate>
```

#### 4.13.1.2 Response

The following example shows the **FolderCreate** command response sent from the server to the client. A <folderhierarchy:Status> value of 1 is returned to indicate that the new folder creation was successful. The <folderhierarchy:SyncKey> value has been incremented in the response, and this value should be used in the next **FolderSync**, **FolderCreate**, **FolderDelete**, or **FolderUpdate** request. The <folderhierarchy:ServerId> value identifies the ID of the new folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderCreate xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>2</SyncKey>
  <ServerId>13</ServerId>
</FolderCreate>
```

### 4.13.2 Deleting Folders by Using the FolderDelete Command

The following examples show how to delete a folder using the **FolderDelete** command.

#### 4.13.2.1 Request

The following example shows how a **FolderDelete** command is used to delete a folder. The <folderhierarchy:SyncKey> element in the request is set to 2, which is the <folderhierarchy:SyncKey> value returned in the last **FolderCreate** response (section 4.13.1.2). The <folderhierarchy:ServerId> value identifies which folder to delete.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderDelete xmlns="FolderHierarchy:">
  <SyncKey>2</SyncKey>
  <ServerId>13</ServerId>
</FolderDelete>
```

#### 4.13.2.2 Response

The following example shows the **FolderDelete** response. A <folderhierarchy:Status> value of 1 is returned to indicate that the folder deletion was successful. The <folderhierarchy:SyncKey> value has been incremented in the response, and this value should be used in the next **FolderSync**, **FolderCreate**, **FolderDelete**, or **FolderUpdate** request.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderDelete xmlns="FolderHierarchy:">
```



Copyright © 2010 Microsoft Corporation.

```
  <Status>1</Status>
  <SyncKey>3</SyncKey>
</FolderDelete>
```

### 4.13.3 Updating Folders by Using the FolderUpdate Command

The following examples show how to update a folder by using the **FolderUpdate** command.

#### 4.13.3.1 Request

The following example shows how a **FolderUpdate** command request is used to update a folder. This request is changing the <folderhierarchy:DisplayName> of the folder to "NewName". The <folderhierarchy:SyncKey> element in the request is set to 3, which is the <folderhierarchy:SyncKey> value returned in the last **FolderSync** response (section 4.13.2.2). The <folderhierarchy:ServerId> identifies the ID of the folder whose name is being updated, and the <folderhierarchy:ParentID> identifies the ID of the parent folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderUpdate xmlns="FolderHierarchy:">
  <SyncKey>3</SyncKey>
  <ServerId>14</ServerId>
  <ParentId>5</ParentId>
  <DisplayName>NewName</DisplayName>
</FolderUpdate>
```

#### 4.13.3.2 Response

The following example shows the **FolderUpdate** command response. A <folderhierarchy:Status> value of 1 is returned to indicate that the folder update was successful. The <folderhierarchy:SyncKey> value has been incremented in the response, and this value should be used in the next **FolderSync**, **FolderCreate**, **FolderDelete**, or **FolderUpdate** request.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderUpdate xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>4</SyncKey>
</FolderUpdate>
```

### 4.13.4 Emptying Folder Contents by Using the ItemOperations Command

The following examples show how to empty the contents of a folder by using the **ItemOperations** command.

#### 4.13.4.1 Request

The following example shows how an **ItemOperations** command request is used to delete the contents of a folder. The <itemoperations:EmptyFolderContents> element contains the <airsync:CollectionId> that identifies the folder whose contents are deleted.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync:" xmlns="ItemOperations:">
  <EmptyFolderContents>
    <airsync:CollectionId>15</airsync:CollectionId>
  </EmptyFolderContents>
</ItemOperations>
```

#### 4.13.4.2 Response

The following example shows the **ItemOperations** command response. An <itemoperations:Status> value of 1 is returned to indicate that the folder deletion was successful. The <itemoperations:EmptyFolderContents> and <airsync:CollectionId> elements provide confirmation of the folder whose contents were deleted.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync:" xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <EmptyFolderContents>
      <Status>1</Status>
      <airsync:CollectionId>15</airsync:CollectionId>
    </EmptyFolderContents>
  </Response>
</ItemOperations>
```

### 4.14 Moving Items to Another Folder by Using the MoveItems Command

The following examples show how to move items to another folder by using the **MoveItems** command.

#### 4.14.1 Request

The following example shows how a **MoveItems** command request is used to move an item from one folder to another. The <move:SrcMsgId> element identifies the item to move. The <move:SrcFldId> element identifies the folder ID of the folder that currently contains the item, and the <move:DstFldId> element identifies the destination folder.

```
<?xml version="1.0" encoding="utf-8"?>
<MoveItems xmlns="Move:">
    <Move>
        <SrcMsgId>5:1</SrcMsgId>
        <SrcFldId>5</SrcFldId>
        <DstFldId>14</DstFldId>
    </Move>
</MoveItems>
```

#### 4.14.2 Response

The following example shows the **MoveItems** command response. A <move:Status> value of 3 is returned to indicate that the move operation was successful. The <move:SrcMsgId> element



Copyright © 2010 Microsoft Corporation.

identifies the original ID of the item to move, and the <move:DstMsgId> element identifies the new ID of the item that was moved.

```
<?xml version="1.0" encoding="utf-8"?>
<MoveItems xmlns="Move:">
  <Response>
    <SrcMsgId>5:1</SrcMsgId>
    <Status>3</Status>
    <DstMsgId>14:1</DstMsgId>
  </Response>
</MoveItems>
```

## 4.15 Creating Meetings

When a user creates an appointment or meeting on the client, the calendar item is added to the server by using the **Sync** command. If the meeting has attendees, the client uses the **SendMail** command to send meeting requests to the attendees. When the attendee synchronizes their Inbox folder, the **Sync** response from the server adds the new meeting request to the attendee's Inbox folder. When the attendee synchronizes their Calendar folder, the **Sync** response from the server adds the new calendar item to the attendee's Calendar folder. This section contains examples that show how a new meeting is uploaded to the server, how the client sends the meeting request, how the new meeting request is added to the attendees Inbox folder, and how the new meeting is added to the attendees Calendar folder.

The **Sync** request command required for creating an appointment is the same as creating a meeting, the only difference is that the initial calendar item uploaded to the server does not have any attendees.

### 4.15.1 Uploading a Meeting to the Server

The following examples show how to upload a meeting to the server by using the **Sync** command.

#### 4.15.1.1 Request

The following example shows a **Sync** command request for the Calendar folder. The **Sync** request contains a new meeting that has one attendee.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar:" xmlns="AirSync:">
    <Collections>
        <Collection>
            <SyncKey>509910681</SyncKey>
            <CollectionId>1</CollectionId>
            <GetChanges>1</GetChanges>
            <Commands>
                <Add>
                    <ClientId>1</ClientId>
                    <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGM
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAAAxP///w==</calendar:TimeZone>
                        <calendar:DtStamp>20100406T170219Z</calendar:DtStamp>
                        <calendar:StartTime>20100503T090000Z</calendar:StartTime>
```



Copyright © 2010 Microsoft Corporation.

```
                        <calendar:Subject>TestMeeting</calendar:Subject>

<calendar:UID>040000008200E00074C5B7101A82E0080000000036BD76EAAAD5CA010000000000000000001000000
0C45185F686A5D542B20BF2CE2F477D55</calendar:UID>
                        <calendar:Attendees>
                            <calendar:Attendee>
                                <calendar:Email>testuser2@contoso.com</calendar:Email>
                                <calendar:Name>Test User 2</calendar:Name>
                                <calendar:AttendeeStatus>0</calendar:AttendeeStatus>
                                <calendar:AttendeeType>1</calendar:AttendeeType>
                            </calendar:Attendee>
                        </calendar:Attendees>
                        <calendar:Location>My Office</calendar:Location>
                        <calendar:EndTime>20100503T100000Z</calendar:EndTime>
                        <calendar:Sensitivity>0</calendar:Sensitivity>
                        <calendar:BusyStatus>1</calendar:BusyStatus>
                        <calendar:AllDayEvent>0</calendar:AllDayEvent>
                    </ApplicationData>
                </Add>
            </Commands>
        </Collection>
    </Collections>
</Sync>
```

#### 4.15.1.2 Response

The following example shows a **Sync** command response for the Calendar folder. The <airsync:SyncKey> value is incremented and a <airsync:Status> value of 1 is returned to indicate that the **Sync** command was successful. The <airsync:ServerId> value is the ID for the new meeting.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>993351228</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Responses>
        <Add>
          <ClientId>1</ClientId>
          <ServerId>1:3</ServerId>
          <Status>1</Status>
        </Add>
      </Responses>
    </Collection>
  </Collections>
</Sync>
```

### 4.15.2 Sending Meeting Requests

The following examples show how to send a meeting request to an attendee's Inbox folder by using the **SendMail** command.



Copyright © 2010 Microsoft Corporation.

#### 4.15.2.1 Request

The following example shows a **SendMail** command request sent by the meeting organizer's client. The **SendMail** command request contains the meeting request for the new calendar item uploaded to the server in section 4.15.1.1.

As specified in [RFC2447] section 3.4, meeting requests have a content type of text/calendar with the *method* parameter set to **REQUEST**. The details of the meeting request are sent in iCalendar format, as specified in [MS-OXCICAL].

```
<?xml version="1.0" encoding="utf-8"?>
<SendMail xmlns="ComposeMail:">
<ClientId>3</ClientId>
<SaveInSentItems/>
<Mime>X-MimeOLE: Produced By Microsoft Exchange V6.5.7226.0
Received: by contoso.com
    id &lt;01C4F431.619431CA@contoso.com&gt;; Tue, 6 Apr 2010 14:52:31 -0800
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="----_=_NextPart_001_01C4F431.619431CA"
Content-class: urn:content-classes:calendarmessage
Subject: TestMeeting
Date: Tue, 6 Apr 2010 17:02:19 -0800
Message-ID: &lt;1E1FC9DA1767ED44872F4E17AC17E8F7011CF265@contoso.com&gt;
X-MS-Has-Attach:
X-MS-TNEF-Correlator:
Thread-Topic: dD)g_&lt;X`WS
Thread-Index: AcT0MWkdH8lvfizmRrysZICO2nAkVQAAAAFg
From: testuser1@contoso.com
To: testuser2@contoso.com

This is a multi-part message in MIME format.

------_=_NextPart_001_01C4F431.619431CA
Content-Type: text/html;
    charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN"&gt;
&lt;HTML&gt;
&lt;HEAD&gt;
&lt;META HTTP-EQUIV=3D"Content-Type" CONTENT=3D"text/html; =
charset=3Diso-8859-1"&gt;
&lt;META NAME=3D"Generator" CONTENT=3D"MS Exchange Server version =
6.5.7232.36"&gt;
&lt;TITLE&gt;TestMeeting&lt;/TITLE&gt;
&lt;/HEAD&gt;
&lt;BODY&gt;
&lt;!-- Converted from text/rtf format --&gt;

&lt;P&gt;&lt;B&gt;&lt;FONT SIZE=3D2 FACE=3D"System"&gt;This is a meeting
request.&lt;/FONT&gt;&lt;/B&gt;
&lt;/P&gt;

&lt;/BODY&gt;
&lt;/HTML&gt;
------_=_NextPart_001_01C4F431.619431CA
```



Copyright © 2010 Microsoft Corporation.

```
Content-class: urn:content-classes:calendarmessage
Content-Type: text/calendar;
    method=REQUEST;
    name="meeting.ics"
Content-Transfer-Encoding: 8bit

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:Microsoft CDO for Microsoft Exchange
VERSION:14.1
BEGIN:VTIMEZONE
TZID:(GMT-08.00) Pacific Time (US &amp;amp; Canada)/Tijuana
X-MICROSOFT-CDO-TZID:13
BEGIN:STANDARD
DTSTART:16010101T020000
TZOFFSETFROM:-0700
TZOFFSETTO:-0800
RRULE:FREQ=YEARLY;WKST=MO;INTERVAL=1;BYMONTH=11;BYDAY=1SU
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:16010101T020000
TZOFFSETFROM:-0800
TZOFFSETTO:-0700
RRULE:FREQ=YEARLY;WKST=MO;INTERVAL=1;BYMONTH=3;BYDAY=2SU
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTAMP:20100406T170219
DTSTART;TZID="(GMT-08.00) Pacific Time (US &amp;amp; Canada)/Tijuana":20100503T090000
SUMMARY:TestMeeting
UID:140000008200E00074C5B7101A82E00800000000E03FFF5AEEF3C401000000000000000000
 0100000000972B1A80D193D54E8DD185652818A128
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-
ACTION;RSVP=TRUE;CN="testuser2":MAILTO:testuser2@contoso.com
ORGANIZER;CN="testuser1":MAILTO:testuser1@contoso.com
LOCATION:My Office\N
DTEND;TZID="(GMT-08.00) Pacific Time (US &amp;amp; Canada)/Tijuana":20100503T100000
DESCRIPTION:Test meeting\N
SEQUENCE:0
PRIORITY:5
CLASS:
CREATED:20100406T170219
LAST-MODIFIED:20100406T170219Z
STATUS:CONFIRMED
TRANSP:OPAQUE
X-MICROSOFT-CDO-BUSYSTATUS:BUSY
X-MICROSOFT-CDO-INSTTYPE:0
X-MICROSOFT-CDO-INTENDEDSTATUS:BUSY
X-MICROSOFT-CDO-ALLDAYEVENT:FALSE
X-MICROSOFT-CDO-IMPORTANCE:1
X-MICROSOFT-CDO-OWNERAPPTID:-2673127979
BEGIN:VALARM
ACTION:DISPLAY
DESCRIPTION:REMINDER
TRIGGER;RELATED=START:-PT00H15M00S
END:VALARM
END:VEVENT
END:VCALENDAR
```



Copyright © 2010 Microsoft Corporation.

```
------_=_NextPart_001_01C4F431.619431CA--</Mime>
</SendMail>
```

#### 4.15.2.2 Response

The following example shows a successful **SendMail** command response. The **SendMail** command response has no XML body (Content-Length: 0) when the **SendMail** command completes successfully.

```
HTTP/1.1 200 OK
Date: Tue, 06 Apr 2010 19:11:25 GMT
Content-Length: 0
```

### 4.15.3 Adding a Meeting Request to the Attendee's Inbox Folder

The following examples show how a meeting request is added to an attendee's Inbox folder using the **Sync** command.

#### 4.15.3.1 Request

The following example shows a **Sync** command request for the Inbox folder of the attendee.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:rm="RightsManagement:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>52305585</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Options>
        <rm:RightsManagementSupport>0</rm:RightsManagementSupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

#### 4.15.3.2 Response

The following example shows a **Sync** command response for the Inbox folder of the attendee. The <airsync:SyncKey> value is incremented and a <airsync:Status> value of 1 is returned to indicate that the **Sync** command was successful. The meeting request is added to the attendees Inbox folder using the contents of the <airsync:ApplicationData> element.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email:" xmlns:email2="Email2:" xmlns:airsyncbase="AirSyncBase:"
xmlns="AirSync:">
  <Collections>
    <Collection>
```



Copyright © 2010 Microsoft Corporation.

```
        <SyncKey>1744744472</SyncKey>
        <CollectionId>5</CollectionId>
        <Status>1</Status>
        <Commands>
          <Add>
            <ServerId>5:1</ServerId>
            <ApplicationData>
              <email:To>"Test User 2" &lt;testuser2@contoso.com&gt;</email:To>
              <email:From>"Test User 1" &lt;testuser1@contoso.com&gt;</email:From>
              <email:Subject>TestMeeting</email:Subject>
              <email:DateReceived>2010-04-06T17:02:26.332Z</email:DateReceived>
              <email:DisplayTo>Test User 2</email:DisplayTo>
              <email:ThreadTopic>TestMeeting</email:ThreadTopic>
              <email:Importance>1</email:Importance>
              <email:Read>0</email:Read>
              <airsyncbase:Body>
                <airsyncbase:Type>3</airsyncbase:Type>
                <airsyncbase:EstimatedDataSize>432</airsyncbase:EstimatedDataSize>
                <airsyncbase:Truncated>1</airsyncbase:Truncated>
              </airsyncbase:Body>
              <email:MessageClass>IPM.Schedule.Meeting.Request</email:MessageClass>
              <email:MeetingRequest>
                <email:AllDayEvent>0</email:AllDayEvent>
                <email:StartTime>2010-05-03T16:00:00.000Z</email:StartTime>
                <email:DtStamp>2010-04-06T17:02:19.499Z</email:DtStamp>
                <email:EndTime>2010-05-03T17:00:00.000Z</email:EndTime>
                <email:InstanceType>0</email:InstanceType>
                <email:Location>My Office</email:Location>
                <email:Organizer>"Test User1" &lt;testuser1@contoso.com&gt;</email:Organizer>
                <email:Reminder>900</email:Reminder>
                <email:ResponseRequested>1</email:ResponseRequested>
                <email:Sensitivity>0</email:Sensitivity>
                <email:BusyStatus>2</email:BusyStatus>

<email:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKABVA
FMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIA
BUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAAxP///w==</email:TimeZone>

<email:GlobalObjId>BAAAAIIA4AB0xbcQGoLgCAAAAAASLNXrHbfKAQAAAAAAAAAAEAAAAMqTtDRC5TFKivAtfhM9F+
8=</email:GlobalObjId>
                <email2:MeetingMessageType>1</email2:MeetingMessageType>
              </email:MeetingRequest>
              <email:InternetCPID>28591</email:InternetCPID>
              <email:Flag />
              <email:ContentClass>urn:content-classes:calendarmessage</email:ContentClass>
              <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
              <email2:ConversationId>…</email2:ConversationId>
              <email2:ConversationIndex>…</email2:ConversationIndex>
              <email:Categories />
            </ApplicationData>
          </Add>
        </Commands>
      </Collection>
    </Collections>
  </Sync>
```



Copyright © 2010 Microsoft Corporation.

### 4.15.4 Adding a Meeting to the Attendee's Calendar Folder

The following examples show how a meeting is added to an attendee's Calendar folder using the **Sync** command.

#### 4.15.4.1 Request

The following example shows a **Sync** command request for the Calendar folder of the **attendee**.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:rm="RightsManagement:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>151676483</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Options>
        <rm:RightsManagementSupport>0</rm:RightsManagementSupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

#### 4.15.4.2 Response

The following example shows a **Sync** command response for the Calendar folder of the attendee. The <airsync:SyncKey> value is incremented and a <airsync:Status> value of 1 is returned to indicate that the **Sync** command was successful. The meeting is added to the attendees Calendar folder using the contents of the <airsync:ApplicationData> element. Because the attendee has not responded to the **meeting request** yet, the <calendar:ResponseType> value is 5, indicating that the meeting request has not been responded to. When the user accepts the meeting request and then synchronizes the Calendar folder again, the <calendar:ResponseType> value will be changed to 3, to indicate that the meeting was accepted.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar:" xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1761460822</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>1:1</ServerId>
          <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGM
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAAAxP///w==</calendar:TimeZone>
            <calendar:DtStamp>20100406T170219Z</calendar:DtStamp>
            <calendar:StartTime>20100503T090000Z</calendar:StartTime>
            <calendar:Subject>TestMeeting</calendar:Subject>
```



Copyright © 2010 Microsoft Corporation.

```
<calendar:UID>040000008200E00074C5B7101A82E00800000000122CD5EB1DB7CA0100000000000000000010000000
0CA93B43442E5314A8AF02D7E133D17EF</calendar:UID>
            <calendar:OrganizerName>Test User 1</calendar:OrganizerName>
            <calendar:OrganizerEmail>testuser1@contoso.com</calendar:OrganizerEmail>
            <calendar:Attendees>
              <calendar:Attendee>
                <calendar:Email>testuser2@contoso.com</calendar:Email>
                <calendar:Name>Test User 2</calendar:Name>
                <calendar:AttendeeType>1</calendar:AttendeeType>
              </calendar:Attendee>
            </calendar:Attendees>
            <calendar:Location>My Office</calendar:Location>
            <calendar:EndTime>20100503T100000Z</calendar:EndTime>
            <airsyncbase:Body>
              <airsyncbase:Type>3</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>275</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <calendar:Sensitivity>0</calendar:Sensitivity>
            <calendar:BusyStatus>1</calendar:BusyStatus>
            <calendar:AllDayEvent>0</calendar:AllDayEvent>
            <calendar:Reminder>15</calendar:Reminder>
            <calendar:MeetingStatus>3</calendar:MeetingStatus>
            <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
            <calendar:ResponseRequested>1</calendar:ResponseRequested>
            <calendar:ResponseType>5</calendar:ResponseType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

## 4.16 Responding to Meeting Requests by Using the MeetingResponse Command

The following example shows how to respond to an exception or single instance of a recurring meeting by using the **MeetingResponse** command and the <meetingresponse:InstanceId> element.

### 4.16.1 Request

The request from the client identifies two instances of recurring meetings that the client is responding to. Note that the first meeting is identified using the <meetingresponse:CollectionId>, <meetingresponse:RequestId>, and <meetingresponse:InstanceId> elements, while the second meeting is identified using the <search:LongId> and <meetingresponse:InstanceId> elements.

```
<?xml version="1.0" encoding="utf-8"?>
<MeetingResponse xmlns:search="Search:" xmlns="MeetingResponse:">
  <Request>
    <UserResponse>2</UserResponse>
    <CollectionId>19</CollectionId>
    <RequestId>19:39</RequestId>
    <InstanceId>20090907T07:00:00.000Z</InstanceId>
  </Request>
```



Copyright © 2010 Microsoft Corporation.

```
  <Request>
    <UserResponse>1</UserResponse>
    <search:LongId>RgAAAju…  …ALS8pQAARL</search:LongId>
    <InstanceId>20091210T17:30:00.000Z</InstanceId>
  </Request>
</MeetingResponse>
```

### 4.16.2 Response

The response from the server includes the <meetingresponse:RequestId> element that was included in the request message, as well as the <meetingresponse:CalendarId> element. The response message is the same whether the <meetingresponse:InstanceId> element is included in the request or not.

```
<?xml version="1.0" encoding="utf-8"?>
<MeetingResponse xmlns:search="Search:" xmlns="MeetingResponse:">
  <Result>
    <RequestId>19:39</RequestId>
    <Status>1</Status>
    <CalendarId>19:43</CalendarId>
  </Result>
  <Result>
    <search:LongId>RgAAAju…  …ALS8pQAARL</search:LongId>
    <Status>1</Status>
    <CalendarId>19:47</CalendarId>
  </Result>
</MeetingResponse>
```

## 4.17 Resolving Recipients and Retrieving Free/Busy Data by Using the ResolveRecipients Command

This section provides sample messages related to the **ResolveRecipients** command.

### 4.17.1 Request

The following example shows how a **ResolveRecipients** command is used to retrieve GAL and contact information using ANR. In this example, the client sends a request to retrieve recipient information for recipients containing the word "testers".

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients:">
  <To>Testers</To>
  <Options>
    <CertificateRetrieval>3</CertificateRetrieval>
    <MaxCertificates>99</MaxCertificates>
    <MaxAmbiguousRecipients>99</MaxAmbiguousRecipients>
  </Options>
</ResolveRecipients>
```



Copyright © 2010 Microsoft Corporation.

### 4.17.2 Response for a GAL Entry

The following example shows two recipients that are being returned to the client. In the "Testers" distribution list, there are three recipients but only two have valid certificates. The value of the <Type> element indicates that the recipient is a GAL entry.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients:">
  <Status>1</Status>
  <Response>
    <To>Testers</To>
    <Status>1</Status>
    <RecipientCount>2</RecipientCount>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Testers</DisplayName>
      <EmailAddress>testers@example.com</EmailAddress>
      <Certificates>
        <Status>1</Status>
        <CertificateCount>2</CertificateCount>
        <RecipientCount>3</RecipientCount>
        <MiniCertificate>AAAAAEfXfBA=</MiniCertificate>
      </Certificates>
    </Recipient>
    <Recipient>
     ...
    </Recipient>
  </Response>
<ResolveRecipients>
```

### 4.17.3 Response for a Contact Entry

The following example shows a response for a contact entry. The value of the <Type> element indicates that the recipient is a contact entry.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients:">
  <Status>1</Status>
  <Response>
    <To>Contact</To>
    <Status>1</Status>
    <RecipientCount>1</RecipientCount>
    <Recipient>
      <Type>2</Type>
      <DisplayName>John Smith</DisplayName>
      <EmailAddress>jsmith@example.com</EmailAddress>
    </Recipient>
  </Response>
</ResolveRecipients>
```

### 4.17.4 Retrieving Free/Busy Data By Using the ResolveRecipients Command

The following examples show a sample request and response in which the free/busy data for two users and two distribution lists are retrieved.



Copyright © 2010 Microsoft Corporation.

### 4.17.4.1 Request to Retrieve Free/Busy Data

The following example shows the request used to resolve two recipients and two distribution lists and retrieve their free/busy information for a two day period.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients:">
  <To>all@contoso.com</To>
  <To>ryan@contoso.com</To>
  <To>Tom</To>
  <To>myPersonalDistributionList</To>
  <Options>
    <MaxAmbiguousRecipients>2</MaxAmbiguousRecipients>
    <Availability>
        <StartTime>2008-12-01T08:00:00.000Z</StartTime>
        <EndTime>2008-12-03T08:00:00.000Z</EndTime>
    </Availability>
  </Options>
</ResolveRecipients>
```

### 4.17.4.2 Response with MergedFreeBusy Data

The following example shows the **ResolveRecipients** response issued for the request in section 4.17.4.1.

As shown in the example, the free/busy data for the all@contoso.com distribution list could not be retrieved and <resolverecipients:Status> value 162 was returned. The free/busy data for Ryan Calafato was returned successfully. Two ambiguous recipient suggestions were returned for "Tom", neither of which contain the <resolverecipients:Availability> element, as it is returned only when an exact match is found. And, the personal distribution list returned a variety of successful and non-successful queries.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients:">
  <Status>1</Status>
  <Response>
    <To>all@contoso.com</To>
    <Status>1</Status>
    <RecipientCount>1</RecipientCount>
    <Recipient>
      <Type>1</Type>
      <DisplayName>All Contoso Full Time Employees</DisplayName>
      <EmailAddress>all@contoso.com</EmailAddress>
      <Availability>
        <Status>162</Status>
      </Availability>
    </Recipient>
  </Response>
  <Response>
    <To>ryan@contoso.com</To>
    <Status>1</Status>
    <RecipientCount>1</RecipientCount>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Ryan Calafato</DisplayName>
```



Copyright © 2010 Microsoft Corporation.

```
      <EmailAddress>ryan@contoso.com</EmailAddress>
      <Availability>
        <Status>1</Status>
        <MergedFreeBusy>00200000000000000000000000100200220000010000000</MergedFreeBusy>
      </Availability>
    </Recipient>
  </Response>
  <Response>
    <To>tom</To>
    <Status>3</Status>
    <RecipientCount>30</RecipientCount>
    <Recipient>
      <Type>2</Type>
      <DisplayName>Tom Getzinger </DisplayName>
      <EmailAddress>tomget@contoso.com</EmailAddress>
    </Recipient>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Tom Higginbotham (Sr.)</DisplayName>
      <EmailAddress>tomhig@contoso.com</EmailAddress>
    </Recipient>
  </Response>
  <Response>
    <To>myPersonalDistributionList</To>
    <Status>1</Status>
    <RecipientCount>4</RecipientCount>
    <Recipient>
      <Type>2</Type>
      <DisplayName>glen@adventureworks.com</DisplayName>
      <EmailAddress>glen@adventureworks.com</EmailAddress>
      <Availability>
        <Status>162</Status>
      </Availability>
    </Recipient>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Tom Higginbotham (Sr.)</DisplayName>
      <EmailAddress>tomhig@contoso.com</EmailAddress>
      <Availability>
        <Status>161</Status>
      </Availability>
    </Recipient>
    <Recipient>
      <Type>2</Type>
      <DisplayName>Steve Riley</DisplayName>
      <EmailAddress>steve@contoso.com</EmailAddress>
      <Availability>
        <Status>1</Status>
        <MergedFreeBusy>33333333333333333333000000100200220000010000000</MergedFreeBusy>
      </Availability>
    </Recipient>
    <Recipient>
      <Type>2</Type>
      <DisplayName>bonnie@adventureworks.com</DisplayName>
      <EmailAddress>bonnie@adventureworks.com</EmailAddress>
      <Availability>
        <Status>162</Status>
      </Availability>
    </Recipient>
```



Copyright © 2010 Microsoft Corporation.

```
  </Response>
</ResolveRecipients>
```

## 4.18 Retrieving and Changing OOF Settings by Using the Settings Command

This section provides sample messages related to retrieving and changing OOF settings.

### 4.18.1 Retrieving OOF Settings

The client requests the user's OOF settings by using the <settings:Get> element and specifying the type in which the client wants to have the OOF message formatted.

#### 4.18.1.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Oof>
    <Get>
      <BodyType>HTML</BodyType>
    </Get>
  </Oof>
</Settings>
```

The client requested the messages to be returned in HTML, so all OOF messages are formatted as such.

#### 4.18.1.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
<Status>1</Status>
  <Oof>
    <Status>1</Status>
    <Get>
      <OofState>2</OofState>
      <StartTime>2007-05-08T10:45:51.250Z</StartTime>
      <EndTime>2007-05-11T10:45:51.250Z</EndTime>
      <OofMessage>
        <AppliesToInternal />
        <Enabled>1</Enabled>
        <ReplyMessage>Internal OOF Message</ReplyMessage>
       <BodyType>HTML</BodyType>
      </OofMessage>
      <OofMessage>
        <AppliesToExternalKnown />
        <Enabled>1</Enabled>
        <ReplyMessage>External OOF Message</ReplyMessage>
         <BodyType>HTML</BodyType>
       </OofMessage>
       <OofMessage>
       <AppliesToExternalUnknown /><Enabled>0</Enabled>
       <ReplyMessage>External OOF Message</ReplyMessage>
       <BodyType>HTML</BodyType>
     </OofMessage>
```



Copyright © 2010 Microsoft Corporation.

```
    </Get>
  </Oof>
</Settings>
```

### 4.18.2 Turning On the OOF Message

The client requires that the OOF message is turned on. The client has to update the OOF status by using the <settings:Set> element.

#### 4.18.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Oof>
    <Set>
      <OofState>2</OofState>
      <OofMessage>
        <AppliesToInternal/>
        <Enabled>1</Enabled>
        <ReplyMessage> &lt;html&gt;&lt;head&gt;&lt;meta
        http-equiv="Content-Type" content="text/html;
        charset=utf-8"&gt;&lt;style&gt;@font-face
        {font-family:Verdana}p.MsoNormal, li.MsoNormal,
        div.MsoNormal {margin:0in; margin-bottom:.0001pt;
        font-size:10.0pt; font-family:Verdana} a:link,
        span.MsoHyperlink {color:blue; text-
        decoration:underline}a:visited,
        span.MsoHyperlinkFollowed {color:purple;
        text-decoration:underline} span.EmailStyle17
        {font-family:Arial; color:windowtext} @page Section1
        {margin:1.0in 1.25in 1.0in 1.25in} div.Section1 {}
        &lt;/style&gt; &lt;/head&gt; &lt;body lang="EN-US"
        link="blue" vlink="purple"&gt; &lt;div class="Section1"&gt;
        &lt;p class="MsoNormal"&gt;&lt;font size="2"
        face="Arial"&gt;&lt;span style="font-size:10.0pt;
        font-family:Arial"&gt;I'll be out of the office
        today.&lt;/span&gt;&lt;/font&gt;&lt;/p&gt; &lt;/div&gt;
        &lt;/body&gt;&lt;/html&gt;</ReplyMessage>
        <BodyType>HTML</BodyType>
      </OofMessage>
      <OofMessage>
        <AppliesToExternalKnown/>
        <Enabled>0</Enabled>
      </OofMessage>
      <OofMessage>
        <AppliesToExternalUnknown/>
        <Enabled>0</Enabled>
      </OofMessage>
    </Set>
  </Oof>
</Settings>
```

#### 4.18.2.2 Response

The server responds with status, to indicate that OOF was successfully enabled.



Copyright © 2010 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

### 4.18.3 Turning Off the OOF Message

The client wants to turn off the OOF message. The client has to update the OOF status by using the <settings:Set> element.

#### 4.18.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Oof>
    <Set>
      <OofState>0</OofState>
    </Set>
  </Oof>
</Settings>
```

#### 4.18.3.2 Response

The server responds with status, to indicate that OOF was successfully disabled.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

## 4.19 Validating Certificates by Using the ValidateCert Command

The following examples show how to validate certificates by using the **ValidateCert** command.

### 4.19.1 Request

The following example shows how the **ValidateCert** command request is used to validate certificates. The <validatecert:CertificateChain> element contains all of the certificates in a certificate chain, and the <validatecert:Certificate> elements contain the individual certificate values. The <validatecert:CheckCrl> element is set to 1 (TRUE), indicating that the server cannot ignore an unverifiable revocation status.

```
<?xml version="1.0" encoding="utf-8"?>
<ValidateCert xmlns="ValidateCert:">
```



Copyright © 2010 Microsoft Corporation.

```
  <CertificateChain>
  <!—Certificate values have been truncated for example purposes -->
    <Certificate>MIICYjCCAcugAwIBAgIUYGs8jZbX0Vxj/0CIrh8…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUdhWamYEKM9eaFVFSylR…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIU9uwT6UARSuw1KdJmYN6…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUB0959dCBM5WSLg7NuM4…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUGFjVCBrvrguSaNxziWN…</Certificate>
  </CertificateChain>
  <Certificates>
    <Certificate>MIICYjCCAcugAwIBAgIUYGs8jZbX0VxjObu4nw0…</Certificate>
  </Certificates>
  <CheckCrl>1</CheckCrl>
</ValidateCert>
```

### 4.19.2 Response

The following example shows the **ValidateCert** command response. A <validatecert:Status> value of 1 is returned to indicate that the certificate validation was successful.

```
<?xml version="1.0" encoding="utf-8"?>
<ValidateCert xmlns="ValidateCert:">
    <Status>1</Status>
    <Certificate>
        <Status>1</Status>
    </Certificate>
</ValidateCert>
```

## 4.20 Retrieving User Information by Using the Settings Command

The following example shows a user-information request and response.

### 4.20.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <UserInformation>
        <Get/>
    </UserInformation>
</Settings>
```

### 4.20.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <Status>1</Status>
    <UserInformation>
        <Status>1</Status>
        <Get>
            <EmailAddresses>
                <SMTPAddress>stefan@contoso.com</SMTPAddress>
            </EmailAddresses>
        </Get>
```



Copyright © 2010 Microsoft Corporation.

```
    </UserInformation>
</Settings>
```

## 4.21 Setting a Device Password by Using the Settings Command

The following example shows a device-password request and response.

### 4.21.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <DevicePassword>
        <Set>
            <Password>bar</Password>
        </Set>
    </DevicePassword>
</Settings>
```

### 4.21.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <Status>1</Status>
    <DevicePassword>
        <Set>
            <Status>...</Status>
        </Set>
    </DevicePassword>
</Settings>
```

## 4.22 Accessing Documents on File Shares and URIs by Using the Search and ItemOperations Commands

This section shows how to use the following process to retrieve an item from a Windows® SharePoint® Services or UNC site:

Issue a **Search** command, specifying the link to the folder. The server will return folder/item metadata, specifying the ID, file name, size, creation date, last modified date, whether the item is a folder, and whether the item is hidden. For instructions on completing this task, see section 4.22.1.

Issue the **ItemOperations** command, specifying the ID from the item metadata. For instructions on completing this task, see section 4.22.

In issuing request 2, the following are considerations for the client pertaining to the size of the file to be retrieved:

- Does the client want to have the item content returned inline in the WBXML, or as separate body parts in the HTTP response? Using WBXML might be easier to implement, but might consume more memory on the device, depending on how the response parser on the device is implemented.
- What is the maximum number of bytes of item content that the client wants to have returned in one response? (Successive requests can be used to obtain the remaining content.)



Copyright © 2010 Microsoft Corporation.

The following figure shows the request and response pattern that is used to find and retrieve an item located on a Windows SharePoint Services or UNC site.

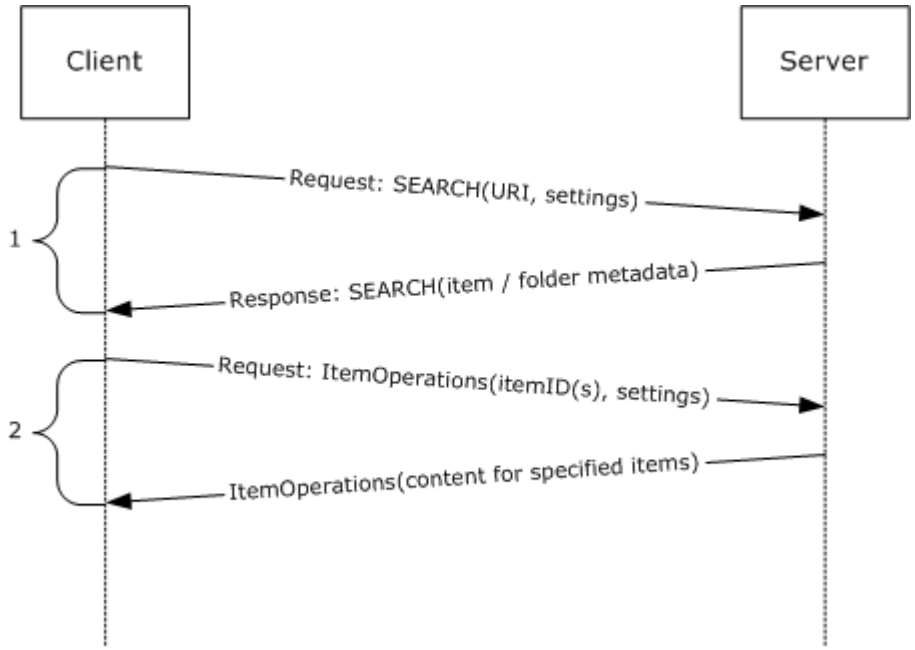


**Figure 8: Finding and retrieving an item from a file share or UNC site**

### 4.22.1 Issuing a Search for Item Metadata

As illustrated in the figure, the client first issues a search request to the server to retrieve metadata about the item (if the URI points to an item) or the items (if the URI points to a folder). The client then does the following:

- Indicates that the client is searching a document library store by using the <Name> element.
- Specifies the URI as the <search:Value> in an <search:EqualTo> query.
- Specifies the range of results that the client wants to have returned in the response.

In this case, the client is attempting to retrieve metadata for the files in a UNC share.

#### 4.22.1.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:documentlibrary="DocumentLibrary:"
xmlns="Search:">
  <Store>
    <Name>DocumentLibrary</Name>
    <Query>
      <EqualTo>
        <documentlibrary:LinkId/>
        <Value>\\somehost\directory</Value>
      </EqualTo>
    </Query>
    <Options>
```



Copyright © 2010 Microsoft Corporation.

```
      <Range>0-999</Range>
    </Options>
  </Store>
</Search>
```

### 4.22.1.2 Response

The response from the server contains the metadata for the folder and items. The very first node in the response is the top-level node, followed by its children (if any).

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:documentlibrary="DocumentLibrary:" xmlns="Search:">
  <Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>directory
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>1
          </documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2007-05-08T17:28:15.375Z
          </documentlibrary:CreationDate>
          <documentlibrary:LastModifiedDate>2007-05-08T17:28:15.406Z
          </documentlibrary:LastModifiedDate>
          <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
        </Properties>
      </Result>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory\resource
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>resource
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>1</documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2004-03-02T12:34:56.123Z
          </documentlibrary:CreationDate>
          <documentlibrary:LastModifiedDate>2005-04-03T12:34:56.345Z
          </documentlibrary:LastModifiedDate>
          <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
        </Properties>
      </Result>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory\TestFile.txt
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>TestFile.txt
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>0</documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2004-03-02T12:34:56.123Z
          </documentlibrary:CreationDate>
          <documentlibrary:LastModifiedDate>2005-04-03T12:34:56.345Z
          </documentlibrary:LastModifiedDate>
          <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
```



Copyright © 2010 Microsoft Corporation.

```
            <documentlibrary:ContentLength>88
            </documentlibrary:ContentLength>
            <documentlibrary:ContentType>text/plain
            </documentlibrary:ContentType>
          </Properties>
        </Result>
        <Range>0-2</Range>
        <Total>3</Total>
      </Store>
    </Response>
  </Search>
```

### 4.22.2 Fetching an Item Based on Metadata

When a document library is used to provide item or folder metadata, the client can retrieve a file within a document library by using the **ItemOperations** command and specifying the <documentlibrary:LinkId> of the item. In this example, the client also specifies that the client only requires bytes from 10 through 19 of the item returned in this request.

#### 4.22.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:documentlibrary="DocumentLibrary:"
xmlns="ItemOperations:">
  <Fetch>
    <Store>DocumentLibrary</Store>
    <documentlibrary:LinkId>\\somehost\directory\
    ActiveSyncDocumentFetch.txt</documentlibrary:LinkId>
    <Options>
      <Range>10-19</Range>
    </Options>
  </Fetch>
</ItemOperations>
```

#### 4.22.2.2 Response

The response from the server contains the requested item. The binary content of the file is base64 encoded and is included in the <itemoperations:Data> element.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:documentlibrary="DocumentLibrary:"
xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <documentlibrary:LinkId>\\somehost\directory\
      ActiveSyncDocumentFetch.txt</documentlibrary:LinkId>
      <Properties>
        <Range>10-19</Range>
        <Total>26</Total>
        <Data>S0xNTk9QUVJTVA==</Data>
        <Version>2005-04-03T12:34:56.345Z</Version>
      </Properties>
    </Fetch>
```



Copyright © 2010 Microsoft Corporation.

```
    </Response>
  </ItemOperations>
```

## 4.23 Using the Supported Element and Ghosted Elements in the Sync Command

This section provides sample messages related to ghosted contact elements and use of the <Supported> element. Many elements in the Contact and Calendar class can be ghosted, as specified in [MS-ASCNTC] and [MS-ASCAL]. When a property is ghosted, its value is retained on the server even when the client sends a **Sync** request with a <airsync:Change> block for only a subset of the class elements. Values for non-ghosted elements are deleted from the server if a value is not specified in the **Sync** request <airsync:Change> block.

The <airsync:Supported> element is included in **Sync** command requests to inform the server that the client is only keeping track of the elements included as children of the <airsync:Supported> element and is not tracking the values of the rest of the class elements.

The example in this section shows the communication between the client and server when the <airsync:Supported> element is used, when the client makes changes to the <airsync:Supported> elements, and when the server makes changes to the Contacts class.

### 4.23.1 Initial Folder Sync

The following examples show the initial **FolderSync** command request and response. The **FolderSync** request uses a <folderhierarchy:SyncKey> value of "0" to indicate an initial synchronization. The **FolderSync** response includes information to populate the user folders on the client device: the folder <folderhierarchy:DisplayName> values, <folderhierarchy:ServerId> values, parent folder (<folderhierarchy:ParentId>) values, and folder <folderhierarchy:Type> values.

#### 4.23.1.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <SyncKey>0</SyncKey>
</FolderSync>
```

#### 4.23.1.2 Response

```
<?xml version="1.0" encoding="utf-8"?><FolderSync xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>1</SyncKey>
  <Changes>
    <Count>11</Count>
    <Add>
      <ServerId>1</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Calendar</DisplayName>
      <Type>8</Type>
    </Add>
    <Add>
      <ServerId>2</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Contacts</DisplayName>
      <Type>9</Type>
    </Add>
```



Copyright © 2010 Microsoft Corporation.

```
      <Add>
        <ServerId>3</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Deleted Items</DisplayName>
        <Type>4</Type>
      </Add>
      <Add>
        <ServerId>4</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Drafts</DisplayName>
        <Type>3</Type>
      </Add>
      <Add>
        <ServerId>5</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Inbox</DisplayName>
        <Type>2</Type>
      </Add>
      <Add>
        <ServerId>6</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Journal</DisplayName>
        <Type>11</Type>
      </Add>
      <Add>
        <ServerId>7</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Junk E-Mail</DisplayName>
        <Type>12</Type>
      </Add>
      <Add>
        <ServerId>8</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Notes</DisplayName>
        <Type>10</Type>
      </Add>
      <Add>
        <ServerId>9</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Outbox</DisplayName>
        <Type>6</Type>
      </Add>
      <Add>
        <ServerId>10</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Sent Items</DisplayName>
        <Type>5</Type>
      </Add>
      <Add>
        <ServerId>11</ServerId>
        <ParentId>0</ParentId>
        <DisplayName>Tasks</DisplayName>
        <Type>7</Type>
      </Add>
    </Changes>
  </FolderSync>
```



Copyright © 2010 Microsoft Corporation.

### 4.23.2 Sync Command

The following examples show the initial **Sync** command request and response. The <airsync:Supported> element is included in the request with two child elements, <contacts:JobTitle> and <contacts:Department>, to indicate to the server that these two elements are being tracked by the client. Note that the <airsync:SyncKey> value is set to "0" when the <airsync:Supported> element is included in the request.

The **Sync** command response indicates that the **Sync** request was processed successfully.

#### 4.23.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:contacts="Contacts:">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>2</CollectionId>
      <Supported>
        <contacts:JobTitle/>
        <contacts:Department/>
      </Supported>
    </Collection>
  </Collections>
</Sync>
```

#### 4.23.2.2 Response

```
<?xml version="1.0" encoding="utf-8"?><Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>878266863</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

### 4.23.3 Sync Contacts

The following examples show the **Sync** command request and response for the Contacts class. The request includes the <airsync:CollectionId> value of "2", which corresponds to the contacts folder as created in section 4.23.1.2.

The **Sync** command response indicates that the **Sync** request was processed successfully.

#### 4.23.3.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>878266863</SyncKey>
      <CollectionId>2</CollectionId>
      <DeletesAsMoves/>
```



Copyright © 2010 Microsoft Corporation.

```
      <GetChanges/>
    </Collection>
  </Collections>
</Sync>
```

### 4.23.3.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:contacts="Contacts:" xmlns:contacts2="Contacts2:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>619052475</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>0</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <contacts:WebPage>http://contoso.com</contacts:WebPage>
            <contacts:BusinessCountry>USA</contacts:BusinessCountry>
            <contacts:Department>Executive</contacts:Department>
            <contacts:Email1Address>"president@contoso.com"
&lt;president@contoso.com&gt;</contacts:Email1Address>
            <contacts:FileAs>Hassall, Mark</contacts:FileAs>
            <contacts:FirstName>Mark</contacts:FirstName>
            <contacts:HomeCity>Seattle</contacts:HomeCity>
            <contacts:HomeCountry>USA</contacts:HomeCountry>
            <contacts:HomePhoneNumber>(206) 555-0100</contacts:HomePhoneNumber>
            <contacts:HomePostalCode>98000</contacts:HomePostalCode>
            <contacts:HomeState>WA</contacts:HomeState>
            <contacts:HomeStreet>234 Main Street</contacts:HomeStreet>
            <contacts:BusinessCity>Seattle</contacts:BusinessCity>
            <contacts:MiddleName>I</contacts:MiddleName>
            <contacts:MobilePhoneNumber>(206) 555-0101</contacts:MobilePhoneNumber>
            <contacts:CompanyName>Contoso Inc.</contacts:CompanyName>
            <contacts:BusinessPostalCode>98000</contacts:BusinessPostalCode>
            <contacts:AssistantName>Andy Jacobs</contacts:AssistantName>
            <contacts:AssistantTelephoneNumber>(206) 555-
0102</contacts:AssistantTelephoneNumber>
            <contacts:LastName>Hassall</contacts:LastName>
            <contacts:BusinessState>WA</contacts:BusinessState>
            <contacts:BusinessStreet>123 Main Street</contacts:BusinessStreet>
            <contacts:BusinessPhoneNumber>(206) 555-0103</contacts:BusinessPhoneNumber>
            <contacts:JobTitle>President</contacts:JobTitle>
            <contacts:OfficeLocation>TopFloor</contacts:OfficeLocation>
            <contacts2:ManagerName>Roya Asbari</contacts2:ManagerName>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
```



Copyright © 2010 Microsoft Corporation.

```
</Sync>
```

### 4.23.4 Sync Client Changes

The following example shows the **Sync** command request and response when the client updates the values for the <contacts:JobTitle> and <contacts:Department> elements. Because all the other contact properties are ghosted, they are not deleted when server processes this request.

The **Sync** command response indicates that the **Sync** request was processed successfully.

#### 4.23.4.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:contacts="Contacts:">
  <Collections>
    <Collection>
      <SyncKey>619052475</SyncKey>
      <CollectionId>2</CollectionId>
      <Commands>
        <Change>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <contacts:JobTitle>Sales Manager</contacts:JobTitle>
            <contacts:Department>Marketing</contacts:Department>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

#### 4.23.4.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>716498022</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

### 4.23.5 Sync Server Changes

The following **Sync** command request and response show the **Sync** command request and response when the <contacts:Manager> value is changed and the <contacts:AssistantName> value is deleted on the server.

#### 4.23.5.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
```



Copyright © 2010 Microsoft Corporation.

```
  <Collections>
    <Collection>
      <SyncKey>716498022</SyncKey>
      <CollectionId>2</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
    </Collection>
  </Collections>
</Sync>
```

### 4.23.5.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:contacts="Contacts:" xmlns:contacts2="Contacts2:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>103384063</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>0</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <contacts:WebPage>http://contoso.com</contacts:WebPage>
            <contacts:BusinessCountry>USA</contacts:BusinessCountry>
            <contacts:Department>Marketing</contacts:Department>
            <contacts:Email1Address>"president@contoso.com"
&lt;president@contoso.com&gt;</contacts:Email1Address>
            <contacts:FileAs>Hassall, Mark</contacts:FileAs>
            <contacts:FirstName>Mark</contacts:FirstName>
            <contacts:HomeCity>Seattle</contacts:HomeCity>
            <contacts:HomeCountry>USA</contacts:HomeCountry>
            <contacts:HomePhoneNumber>(206) 555-0100</contacts:HomePhoneNumber>
            <contacts:HomePostalCode>98000</contacts:HomePostalCode>
            <contacts:HomeState>WA</contacts:HomeState>
            <contacts:HomeStreet>234 Main Street</contacts:HomeStreet>
            <contacts:BusinessCity>Seattle</contacts:BusinessCity>
            <contacts:MiddleName>I</contacts:MiddleName>
            <contacts:MobilePhoneNumber>(206) 555-0101</contacts:MobilePhoneNumber>
            <contacts:CompanyName>Contoso Inc.</contacts:CompanyName>
            <contacts:BusinessPostalCode>98000</contacts:BusinessPostalCode>
            <contacts:AssistantTelephoneNumber>(206) 555-
0102</contacts:AssistantTelephoneNumber>
            <contacts:LastName>Hassall</contacts:LastName>
            <contacts:BusinessState>WA</contacts:BusinessState>
            <contacts:BusinessStreet>123 Main Street</contacts:BusinessStreet>
            <contacts:BusinessPhoneNumber>(206) 555-0103</contacts:BusinessPhoneNumber>
            <contacts:JobTitle>Sales Manager</contacts:JobTitle>
            <contacts:OfficeLocation>TopFloor</contacts:OfficeLocation>
            <contacts2:ManagerName>Carole Poland</contacts2:ManagerName>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          </ApplicationData>
```



Copyright © 2010 Microsoft Corporation.

```
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

## 4.24 Moving a Conversation by Using the ItemOperations Command

The following examples show how to move a conversation to another folder by using the **ItemOperations** command.

### 4.24.1 Request

The following example shows how an **ItemOperations** command request is used to move a conversation from one folder to another. The <itemoperations:ConversationId> element in the request identifies the conversation to move. The <itemoperations:DstFldId> element identifies the destination folder, and the <itemoperations:MoveAlways> element indicates that all future messages that are part of the specified conversation be moved to the destination folder as well.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations:">
  <Move>

    <DstFldId>15</DstFldId>
    <ConversationId>...</ConversationId>
    <Options>
      <MoveAlways/>
    </Options>
  </Move>
</ItemOperations>
```

### 4.24.2 Response

The following example shows the **ItemOperations** command response. An <itemoperations:Status> value of 1 is returned to indicate that the move operation was successful. The <itemoperations:ConversationId> value is returned to confirm the conversation that was moved.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations:">
  <Status>1</Status>
  <Response>
    <Status>1</Status>
    <Move>
      <Status>1</Status>
      <ConversationId>...</ConversationId>
    </Move>
  </Response>
</ItemOperations>
```



Copyright © 2010 Microsoft Corporation.

# 5 Security

## 5.1 Security Considerations for Implementers

The device honors all policies sent down by the server, or sends up the appropriate status codes indicating the non-success.

## 5.2 Index of Security Parameters

| Security Parameter | Section |
|---|---|
| Provision Command | 2.2.2.12 |
| ValidateCert Command | 2.2.2.20 |



Copyright © 2010 Microsoft Corporation.

# 6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products:

- Microsoft® Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft® Exchange Server 2007 Service Pack 3 (SP3)
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 2010 Service Pack 1 (SP1)

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. If a product edition appears with the product version, behavior is different in that product edition.

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 2.2.2.1: When sending an **Autodiscover** command request to Exchange 2007, the Content-Type header accepts the following values: text/html or text/xml.

<2> Section 2.2.2.1.2.1.1.1: In Exchange 2007, the <Culture> element always returns "en:en", regardless of the culture that is sent by the client.

<3> Section 2.2.2.6: The **GetAttachment** command is not supported when the MS-ASProtocolVersion header is set to 14.0 or 14.1 in the **GetAttachment** command request. Use the <Fetch> element of the **ItemOperations** command instead. For more information about the MS-ASProtocolVersion header, see [MS-ASHTTP] section 2.2.2.1.2.1. For more information about the applicability of the MS-ASProtocolVersion header value to Exchange 2007 and Exchange 2010, see [MS-ASHTTP] section 1.6.

<4> Section 2.2.2.7.1.1.1.1: The <FilterType> element is a supported child element of <Collection> in requests when the MS-ASProtocolVersion header is set to 12.1.

<5> Section 2.2.2.7.1.1.1.1.1: When the MS-ASProtocolVersion header value is set to 12.1, the <airsync:SyncKey> element is placed after the <FilterType> element in a **GetItemEstimate** command request.

<6> Section 2.2.2.7.1.1.1.1.3: The <airsync:ConversationMode> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<7> Section 2.2.2.7.1.1.1.1.4: The <airsync:Options> element is not supported in a **GetItemEstimate** request when the MS-ASProtocolVersion header is set to 12.1.

<8> Section 2.2.2.7.1.1.1.1.4.1: The <airsync:Class> element not supported as a child of the <airsync:Options> element when the MS-ASProtocolVersion header is set to 12.1. If the MS-ASProtocolVersion is set to 12.1, the <airsync:Class> element is a child of the <Collection> block.

<9> Section 2.2.2.7.1.1.1.1.4.2: The <Collection> element is a supported parent element of <FilterType> when the MS-ASProtocolVersion header is set to 12.1.



Copyright © 2010 Microsoft Corporation.

<10> Section 2.2.2.7.1.1.1.1.4.3: The <airsync:MaxItems> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<11> Section 2.2.2.7.2.1.1.2: The <FilterType> element is a supported child element of <Collection> in requests when the MS-ASProtocolVersion header is set to 12.1.

<12> Section 2.2.2.8.2.1.3: The <Move> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<13> Section 2.2.2.8.2.1.3.1: The <ConversationId> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<14> Section 2.2.2.8.2.1.3.2: The <DstFldId> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<15> Section 2.2.2.8.2.1.3.3.1: The <MoveAlways> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<16> Section 2.2.2.8.3.1.2.1.2: The <ConversationId> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<17> Section 2.2.2.9: In Exchange 2007, the **MeetingResponse** command is used to accept, tentatively accept, or decline a meeting request in the user's Inbox folder only. The Calendar folder cannot be used to modify meeting requests in Exchange 2007.

<18> Section 2.2.2.9.1.1.1.4: The <InstanceId> element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0. A <Status> value of 2 is returned if the <InstanceId> element is included in requests in which the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<19> Section 2.2.2.11.1.1.2.1.2: The *Notes* value is not supported when the MS-ASProtocolVersion header is set to 12.1.

<20> Section 2.2.2.13: Retrieval of free/busy information using the <Availability> element in the **ResolveRecipients** command is not supported when the MS-ASProtocolVersion header is set to 12.1.

<21> Section 2.2.2.13.1.1.1: Some fields that are ANR-indexed in **Active Directory** by default in Exchange 2007 are as follows: Name, Alias, Email, Office. The ANR property set that can be indexed is definable by the administrator and it can be extended to include other fields.

<22> Section 2.2.2.13.1.1.2: The <Picture> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<23> Section 2.2.2.13.1.1.2.4: The <Availability> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<24> Section 2.2.2.13.1.1.2.4.1: The <StartTime> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<25> Section 2.2.2.13.1.1.2.4.2: The <EndTime> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<26> Section 2.2.2.13.1.1.2.5: The <Picture> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<27> Section 2.2.2.13.1.1.2.5.1: The <MaxSize> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.



Copyright © 2010 Microsoft Corporation.

<28> Section 2.2.2.13.1.1.2.5.2: The <MaxPictures> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<29> Section 2.2.2.13.2.1.2.1: Some fields that are ANR-indexed in Active Directory by default are as follows: Name, Alias, Email, Office. The ANR property set that can be indexed is definable by the administrator and can be extended to include other fields.

<30> Section 2.2.2.13.2.1.2.4.4: The <Availability> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<31> Section 2.2.2.13.2.1.2.4.4.1: The <Availability> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<32> Section 2.2.2.13.2.1.2.4.4.2: The <MergedFreeBusy> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<33> Section 2.2.2.13.2.1.2.4.6: The <Picture> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<34> Section 2.2.2.13.2.1.2.4.6.1: The <Picture> element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<35> Section 2.2.2.13.2.1.2.4.6.2: The <Data> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<36> Section 2.2.2.14.1.1.1.2.1.2: The following classes are supported for mailbox searches when the MS-ASProtocolVersion header is set to 12.1: Email, Calendar, Contacts, Tasks. The SMS and Notes classes are only available if the MS-ASProtocolVersion header is set to 14.0 or 14.1.

<37> Section 2.2.2.14.1.1.1.2.1.4: The <ConversationId> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<38> Section 2.2.2.14.1.1.1.3.7: The <Picture> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<39> Section 2.2.2.14.1.1.1.3.7.1: The <MaxSize> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<40> Section 2.2.2.14.1.1.1.3.7.2: The <MaxPictures> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<41> Section 2.2.2.14.2.1.2.1.2.1: The following classes are supported for mailbox searches when the MS-ASProtocolVersion header is set to 12.1: Email, Calendar, Contacts, Tasks. The SMS and Notes classes are only available if the MS-ASProtocolVersion header is set to 14.0 or 14.1.

<42> Section 2.2.2.14.2.1.2.1.2.4.1: The <gal:Picture> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<43> Section 2.2.2.14.2.1.2.1.2.4.1.1: The <gal:Picture> element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<44> Section 2.2.2.14.2.1.2.1.2.4.1.2: The <gal:Data> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<45> Section 2.2.2.15.1.1.2: The <AccountId> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0. Exchange 2007 returns <Status> value 103 if the <AccountId> element is included in a **SendMail** command request.



Copyright © 2010 Microsoft Corporation.

<46> Section 2.2.2.16: Sending the <DeviceInformation> parameters immediately after the client has been provisioned, and before the **FolderSync** command is not recommended for Exchange 2007.

<47> Section 2.2.2.16.1.1.1: The <RightsManagementInformation> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<48> Section 2.2.2.16.1.1.2.2.4: Exchange 2007 and Exchange 2010 require that the reply message for unknown external and known external audiences be the same.

<49> Section 2.2.2.16.1.1.4: When the MS-ASProtocolVersion header value is 12.1 or 14.0, clients SHOULD use the **Settings** command to send <DeviceInformation> parameters to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command, so that the server can use this information to determine what the device has access to.

<50> Section 2.2.2.16.1.1.4.1: Sending the <DeviceInformation> parameters immediately after the client has been provisioned, and before the **FolderSync** command is not recommended for Exchange 2007.

<51> Section 2.2.2.16.1.1.4.1.6: When the <EnableOutboundSMS> element is set to 1 and the MS-ASProtocolVersion header is set to 14.0, the <PhoneNumber> element is required to have a value. Under these conditions, if the <PhoneNumber> element does not have a value, a <Status> value of 5 is returned by the server.

<52> Section 2.2.2.16.1.1.4.1.8: The <EnableOutboundSMS> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<53> Section 2.2.2.16.1.1.4.1.8: When the <EnableOutboundSMS> element is set to 1 and the MS-ASProtocolVersion header is set to 14.0, the <PhoneNumber> element is required to have a value. Under these conditions, if the <PhoneNumber> element does not have a value, a <Status> value of 5 is returned by the server.

<54> Section 2.2.2.16.1.1.4.1.9: The <MobileOperator> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<55> Section 2.2.2.16.2.1.2.2.4: Exchange 2007 requires that the reply message for unknown external and known external audiences be the same.

<56> Section 2.2.2.16.2.1.3: When the MS-ASProtocolVersion header value is 14.0, clients SHOULD send <DeviceInformation> parameters using the **Settings** command to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command, so that the server can use this information to determine what the device has access to.

<57> Section 2.2.2.16.2.1.5.2.1: The <Accounts> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<58> Section 2.2.2.16.2.1.5.2.1.1: The <Account> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<59> Section 2.2.2.16.2.1.5.2.1.1.1: The <AccountId> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<60> Section 2.2.2.16.2.1.5.2.1.1.2: The <AccountName> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<61> Section 2.2.2.16.2.1.5.2.1.1.3: The <UserDisplayName> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.



Copyright © 2010 Microsoft Corporation.

<62> Section 2.2.2.16.2.1.5.2.1.1.4: The <SendDisabled> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<63> Section 2.2.2.16.2.1.5.2.1.1.5.2: The <PrimarySmtpAddress> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<64> Section 2.2.2.16.2.1.6: The <RightsManagementInformation> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<65> Section 2.2.2.17.1.1.3: The <AccountId> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0. Exchange 2007 returns <Status> value 103 if the <AccountId> element is included in a **SmartForward** command request.

<66> Section 2.2.2.18.1.1.3: The <AccountId> element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0. Exchange 2007 returns <Status> value 103 if the <AccountId> is included in a **SmartReply** command request.

<67> Section 2.2.2.19.1.2.1.1.3: The **Sync** command returns a <Status> value of 4 when the <Supported> element is included in a **Sync** request where the MS-ASProtocolVersion is set to 12.1 and the request is sent to an Exchange 2007 SP1 server.

<68> Section 2.2.2.19.1.2.1.1.3: The <ResponseRequested> element is not supported as a child of the <Supported> element when the MS-ASProtocolVersion header value is set to 12.1.

<69> Section 2.2.2.19.1.2.1.1.3: The <DisallowNewTimeProposal> element is not supported as a child of the <Supported> element when the MS-ASProtocolVersion header value is set to 12.1.

<70> Section 2.2.2.19.1.2.1.1.7: The <ConversationMode> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<71> Section 2.2.2.19.1.2.1.1.8.2: The <Class> element is not supported as a child of the <Options> element when the MS-ASProtocolVersion header is set to 12.1. When the MS-ASProtocolVersion is set to 12.1, the <Class> element is a child of the <Collection> element.

<72> Section 2.2.2.19.1.2.1.1.8.6: The <MaxItems> element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<73> Section 2.2.2.19.1.2.1.1.9.3.1: The <Class> element is not supported as a child of the <Add> element when the MS-ASProtocolVersion header is set to 12.1. When the MS-ASProtocolVersion is set to 12.1, the <Class> element is a child of the <Collection> element.

<74> Section 2.2.2.19.2.1.3.1.4.1: The <Delete> and <Class> elements are not returned in the **Sync** response for an SMS deletion when the MS-ASProtocolVersion header is set to 14.0.

<75> Section 2.2.2.19.2.1.3.1.4.1.1: The <Class> element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<76> Section 2.2.2.19.2.1.3.1.4.1.1: The <Delete> and <Class> elements are not returned in the **Sync** response for an SMS deletion when the MS-ASProtocolVersion header is set to 14.0.

<77> Section 2.2.2.19.2.1.3.1.4.3.1: The <Class> element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<78> Section 2.2.2.19.2.1.3.1.4.5.1.1: The <Class> element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<79> Section 2.2.2.19.2.1.3.1.4.5.2.1: The <Class> element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.



Copyright © 2010 Microsoft Corporation.

<80> Section 2.2.2.19.2.1.3.1.5: In Exchange 2007, the server sends **Sync** response messages containing the <MoreAvailable> element and between zero (0) and <WindowSize> schema changes when it encounters elements external to the protocol. If the client receives multiple **Sync** responses that contain the <MoreAvailable> element and fewer changes than requested by the <WindowSize> value included in the **Sync** request, then the client SHOULD continue to send **Sync** requests to ensure that all in-protocol schema changes have been received by the client. If this **Sync** request and response loop is affecting network performance and synchronizing the client is of less importance than network performance, then the client SHOULD stop sending **Sync** requests.

<81> Section 2.2.3: In Exchange 2007, this was an HTTP 400 response.

<82> Section 2.2.3: In Exchange 2007, this was an HTTP 400 response, or 500 for **SendMail**.

<83> Section 2.2.3: In Exchange 2007, this was an HTTP 500 response.

<84> Section 2.2.3: In Exchange 2007, this was an HTTP 503 response.

<85> Section 2.2.3: In Exchange 2007, this was an HTTP 403 response.

<86> Section 2.2.3: In Exchange 2007, this was an HTTP 507 response.

<87> Section 2.2.3: In Exchange 2007, this was an HTTP 500 response, or 403 for **Provision**.

<88> Section 2.2.3: In Exchange 2007, this was an HTTP 501 response.

<89> Section 2.2.3: In Exchange 2007, this was an HTTP 400 response, or 505 for version 1.0 devices.

<90> Section 2.2.3: In Exchange 2007, this was an HTTP 449 response, or 403 if there was no policy key header.

<91> Section 2.2.3: In Exchange 2007, this was an HTTP 449 response.

<92> Section 2.2.3: In Exchange 2007, this was an HTTP 400 or 501 response.

<93> Section 2.2.3: Status value 166 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<94> Section 2.2.3: Status value 167 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<95> Section 2.2.3: Status value 166 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<96> Section 2.2.3: Status value 169 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<97> Section 2.2.3: Status value 170 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<98> Section 2.2.3: Status value 171 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<99> Section 2.2.3: Status value 172 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<100> Section 2.2.3: Status value 173 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.



*Copyright © 2010 Microsoft Corporation.*

<101> Section 2.2.3: Status value 174 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<102> Section 2.2.3: Status value 175 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<103> Section 2.2.3: Status value 176 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<104> Section 2.2.3: Status value 177 is not returned when the MS-ASProtocolVersion header value is set to 12.1. In addition, the initial release version of Exchange 2007 does not return status value 177 when the MS-ASProtocolVersion header value is set to 14.0.

<105> Section 3.1.5.2: Sending the <settings:DeviceInformation> parameters immediately after the client has been provisioned and before the **FolderSync** command is not recommended for Exchange 2007.

<106> Section 4.2.6: In Exchange 2007 and Exchange 2010, the 401-1.htm Web page that is installed in the Help subdirectory of the Autodiscover physical directory can be configured as shown in this section to provide additional troubleshooting details.



Copyright © 2010 Microsoft Corporation.

# 7 Change Tracking

This section identifies changes that were made to the [MS-ASCMD] protocol document between the August 2010 and November 2010 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- Changes made for template compliance.
- Removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.



Copyright © 2010 Microsoft Corporation.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change Type |
|---|---|---|---|
| 1.4 Relationship to Other Protocols | 56370<br>Removed reference to complex types. | N | Content updated. |
| 2.2.2.14 Search | 55501<br>Removed information about the maximum number of Search results returned. | N | Content updated. |
| 2.2.2.14.1.1.1.3.7.2 MaxPictures | 56370<br>Added (request only) to the parent elements column. | N | Content updated. |
| 2.2.2.15.2 Response | 55687<br>Added reference to [XMLSCHEMA1] and changed the word "schema" to "XSD." | N | Content updated. |
| 2.2.2.16.1 Request | 55687<br>Removed minOccurs="1" and maxOccurs="1" from the XSD as they are the default values as specified in [XMLSCHEMA1]. Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.2.17.1 Request | 55687<br>Removed minOccurs="1" and maxOccurs="1" from the XSD as they are the default values as specified in [XMLSCHEMA1]. Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.2.17.2 Response | 55687<br>Added a reference to [XMLSCHEMA1]. | N | Content updated. |



Copyright © 2010 Microsoft Corporation.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change Type |
|---|---|---|---|
| 2.2.2.18.1 Request | 55687<br>Removed minOccurs="1" and maxOccurs="1" from the XSD as they are the default values as specified in [XMLSCHEMA1]. Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.2.18.2 Response | 55687<br>Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.2.19.1.2.1.1.3 Supported | 47887<br>Added a product behavior note stating that the Sync command returns an error when the <Supported> element is sent from a client using MS-ASProtocolVersion 12.1 to an Exchange 2007 SP1 server. | Y | New product behavior note added. |
| 2.2.2.19.1.2.1.1.9.1 Change | 56539<br>Removed reference to the <RTF> element and changed <Picture> to <contacts:Picture>. | N | Content updated. |
| 2.2.2.19.2.1.3.1.4.1.1 Class | 56370<br>Removed information about the <Options>, <Add>, and <Change> parent element as these parent elements and their child <Class> elements are specified in their own section of the specification. | N | Content updated. |
| 2.2.2.20.1 Request | 55687<br>Removed minOccurs="1" and maxOccurs="1" from the XSD as they are the default values as specified in [XMLSCHEMA1]. Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.2.20.2 Response | 55687<br>Removed minOccurs="1" and maxOccurs="1" from the XSD as they are the default values as specified in [XMLSCHEMA1]. Added a reference to [XMLSCHEMA1]. | N | Content updated. |
| 2.2.3 Common Status Codes | 57291<br>Revised descriptions for response codes 106, 110, 112, 123, 124, 125, 126, and 155. | N | Content updated. |
| 2.2.3 Common Status Codes | 58461<br>Updated behavior note for status value 177. | N | Product behavior note updated. |
| 4.5.7.4 Sync Response with MIME Support | 55324<br>Added the Email2 namespace definition, <email2:ConversationId>, and <email2:ConversationIndex> elements to the example code. | N | Content updated. |
| 4.10.2.2 | 55324<br>Added the Email2 namespace definition, the | N | Content |



Copyright © 2010 Microsoft Corporation.

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change Type |
|---|---|---|---|
| Response | <email2:ConversationId>, and <email2:ConversationIndex> elements to the example code. | | updated. |
| 4.10.3.4 Fetch Response | 55324<br>Removed the header, the <airsync:CollectionId>, and <airsync:ServerId> from the example code. Added the Email2 namespace definition, the <email2:ConversationId>, and <email2:ConversationIndex> elements to the example code. | N | Content updated. |



Copyright © 2010 Microsoft Corporation.

# 8 Index


**A**

Applicability 22
Autodiscover message 24

**C**

Capability negotiation 22
Change tracking 335
Client
  abstract data model 254
  higher-layer triggered events 254
  initialization 254
  overview 254
  sequencing rules 254
  timers (section 3.1.2 254, section 3.1.6 260)
Commands message 24

**E**

Examples
  overview 261

**F**

FolderCreate message 33
FolderDelete messagess 38
FolderSync message 42
FolderUpdate message 52

**G**

Glossary 17

**I**

Implementer – security considerations 327
Index of security parameters 327
Informative references 20
Introduction 17

**M**

Messages
  Autodiscover message 24
  Commands 24
  FolderCreate message 33
  FolderDelete message 38
  FolderSync message 42
  FolderUpdate message 52
  overview 23
  syntax 23
  transport 23

**N**

Normative references 19

**O**

Overview (synopsis) 20

**P**

Parameters – security index 327
Preconditions 22
Prerequisites 22
Product behavior 328

**R**

References
  informative 20
  normative 19
Relationship to other protocols 21

**S**

Security
  implementer considerations 327
  overview 327
  parameter index 327
Server
  abstract data model 254
  higher-layer triggered events 254
  initialization 254
  overview 254
  sequencing rules 254
  timers (section 3.1.2 254, section 3.1.6 260)
Standards assignments 22

**T**

Tracking changes 335
Transport 23

**V**

Vendor-extensible fields 22
Versioning 22





*Copyright © 2010 Microsoft Corporation.*