

**Exchange EAS Test Suites Specification**

**Contents**

[1 Introduction 3](#_Toc435694095)

[2 Requirement specification 4](#_Toc435694096)

[3 Design considerations 5](#_Toc435694097)

[3.1 Assumptions 5](#_Toc435694098)

[3.2 Dependencies 5](#_Toc435694099)

[4 Package design 6](#_Toc435694100)

[4.1 Architecture 6](#_Toc435694101)

[4.2 Common library 7](#_Toc435694102)

[4.2.1 ActiveSyncClient 7](#_Toc435694103)

[4.2.2 Helper methods 7](#_Toc435694104)

[4.2.3 Message structures 7](#_Toc435694105)

[4.3 Adapter 7](#_Toc435694106)

[4.3.1 Protocol Adapter 7](#_Toc435694107)

[4.3.2 SUT Control Adapter 7](#_Toc435694108)

[4.4 Test suite 8](#_Toc435694109)

[4.4.1 MS-ASAIRS 8](#_Toc435694110)

[4.4.2 MS-ASCAL 8](#_Toc435694111)

[4.4.3 MS-ASCMD 8](#_Toc435694112)

[4.4.4 MS-ASCNTC 9](#_Toc435694113)

[4.4.5 MS-ASCON 10](#_Toc435694114)

[4.4.6 MS-ASDOC 10](#_Toc435694115)

[4.4.7 MS-ASEMAIL 10](#_Toc435694116)

[4.4.8 MS-ASHTTP 10](#_Toc435694117)

[4.4.9 MS-ASNOTE 11](#_Toc435694118)

[4.4.10 MS-ASPROV 11](#_Toc435694119)

[4.4.11 MS-ASRM 11](#_Toc435694120)

[4.4.12 MS-ASTASK 11](#_Toc435694121)

# Introduction

The Exchange EAS Protocol Test Suites are implemented as synthetic clients running against a server-side implementation of a given Exchange protocol. They are designed in a client-to-server relationship and were originally developed for the in-house testing of the Microsoft Open Specifications. Test suites have been used extensively in Plugfests and Interoperability Labs to test partner implementations.

This document describes how the Exchange EAS Protocol Test Suites are designed to verify that the server behaves in the way that is compliant with normative protocol requirements as described in the technical specification.

The Microsoft Open Specifications were written using the normative language defined in [RFC2119](http://go.microsoft.com/fwlink/?LinkId=117453). The statements of them are extracted as protocol requirements which are listed in the requirement specification described in section 2. The test suites are developed to test the normative protocol requirements. In a single test suite, similar or related requirements are grouped into one test case, and the test cases about same command or operation are grouped into one scenario.

The technical specifications listed in the following table are included in the Exchange EAS Protocol Test Suites package. The version of these technical specifications is v20150630.

Exchange EAS Protocol technical specifications

|  |  |  |
| --- | --- | --- |
| Technical specification | | Protocol name |
| MS-ASAIRS | [Exchange ActiveSync: AirSyncBase Namespace Protocol](http://go.microsoft.com/fwlink/?LinkID=301330) | |
| MS-ASCAL | [Exchange ActiveSync: Calendar Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301331) | |
| MS-ASCMD | [Exchange ActiveSync: Command Reference Protocol](http://go.microsoft.com/fwlink/?LinkID=301332) | |
| MS-ASCNTC | [Exchange ActiveSync: Contact Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301333) | |
| MS-ASCON | [Exchange ActiveSync: Conversations Protocol](http://go.microsoft.com/fwlink/?LinkID=301334) | |
| MS-ASDOC | [Exchange ActiveSync: Document Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301335) | |
| MS-ASDTYPE | [Exchange ActiveSync: Data Types](http://go.microsoft.com/fwlink/?LinkID=708453) | |
| MS-ASEMAIL | [Exchange ActiveSync: Email Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301336) | |
| MS-ASHTTP | [Exchange ActiveSync: HTTP Protocol](http://go.microsoft.com/fwlink/?LinkID=301337) | |
| MS-ASNOTE | [Exchange ActiveSync: Notes Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301338) | |
| MS-ASPROV | [Exchange ActiveSync: Provisioning Protocol](http://go.microsoft.com/fwlink/?LinkID=301339) | |
| MS-ASRM | [Exchange ActiveSync: Rights Management Protocol](http://go.microsoft.com/fwlink/?LinkID=301340) | |
| MS-ASTASK | [Exchange ActiveSync: Tasks Class Protocol](http://go.microsoft.com/fwlink/?LinkID=301341) | |
| MS-ASWBXML | [Exchange ActiveSync: WAP Binary XML (WBXML) Algorithm](http://go.microsoft.com/fwlink/?LinkID=708454) | |

# Requirement specification

A requirement specification contains a list of requirements that are extracted from statements in the technical specification. Each technical specification has one corresponding requirement specification named as MS-XXXX\_RequirementSpecification.xlsx, which can be found in the Docs\MS-XXXX folder in the Exchange EAS Protocol Test Suites package together with the technical specification.

The requirements are categorized as normative or informative. If the statement of the requirement is required for interoperability, the requirement is normative. If the statement of the requirement is clarifying information or high-level introduction, and removal of it does not affect interoperability, the requirement is informative.

Each requirement applies to a specific scope: server, client, or both. If the requirement describes a behavior performed by the responder, the scope of the requirement is server. If the requirement describes a behavior performed by the initiator, the scope of the requirement is client. If the requirement describes a behavior performed by both initiator and responder, the scope of the requirement is both.

The test suites cover normative requirements which describes a behavior performed by the responder. For a detailed requirements list and classification, see the MS-XXXX\_RequirementSpecification.xlsx.

# Design considerations

## Assumptions

* The Exchange EAS Protocol Test Suites are not designed to run multi-protocol user scenarios, but rather provide a way to exercise certain operations documented in a technical specification.
* The test suites are functional tests that verify the compatibility of the system under test (SUT) with a protocol implementation.
* The test suites do not cover every protocol requirement and in no way certify an implementation, even if all tests pass.
* The test suites verify the server-side testable requirements; they do not verify the requirements related to client behaviors and server internal behaviors.
* The test suites assumes that there is a shared folder on the server.
* The test suites assumes that there is a hidden folder and a visible folder under the shared folder.
* The test suites assumes that there is a hidden document and a visible document under the shared folder.
* The test suites assumes that authentication has been performed by the underlying protocols.

## Dependencies

* All Exchange EAS Protocol Test Suites depend on the Protocol Test Framework (PTF) to derive managed adapters.
* All Exchange EAS Protocol Test Suites depends on the HTTP protocol or HTTPS protocol to transmit the messages.
* All Exchange EAS Protocol Test Suites depends on MS-ASWBXML to encode XML requests bodies into WBXML for transmission to an ActiveSync server.
* All Exchange EAS Protocol Test Suites depends on MS-ASHTTP to synchronize the data which is stored on the server.
* All Exchange EAS Protocol Test Suites depends on the xsd.exe tool in the .NET Framework SDK to generate structures used in the MS-ASCAL request and response.

# Package design

The Exchange EAS Protocol Test Suites are implemented as synthetic clients running against a server-side implementation of a given Exchange protocol. The test suites verify the server-side and testable requirements.

## Architecture

The following figure illustrates the Exchange EAS Protocol Test Suites architecture.



**Figure 1: Architecture**

The following outlines the details of the test suites architecture:

**SUT**

The system under test (SUT) hosts the server-side implementation of the protocol, which test suites run against.

* + From a third-party’s point of view, the SUT is a server implementation.
  + The following products have been tested with the test suites on the Windows platform.
* Microsoft Exchange Server 2007 Service Pack 3 (SP3)
* Microsoft Exchange Server 2010 Service Pack 3 (SP3)
* Microsoft Exchange Server 2013 Service Pack 1 (SP1)
* Microsoft Exchange Server 2016

**Test Suite Client**

The test suites act as synthetic clients to communicate with the SUT and validate the requirements gathered from technical specifications. The Exchange EAS Protocol Test Suites include one common library, 13 adapters and 12 test suites.

* The test suites communicate with SUT via a protocol adapter and SUT control adapter to verify if the SUT behaves in the way that is compliant with normative protocol requirements.
* All protocol adapters uses ActiveSyncClient to send command request and retrieve command response.
* ActiveSyncClient encodes and decodes commands defined in [MS-ASCMD] by using MS-ASWBXML and communicates with the SUT via MS-ASHTTP.

## Common library

The common library provides implementation of the ActiveSyncClient, messages, structures, and helper methods.

### ActiveSyncClient

The ActiveSyncClient works as an intermediary between the protocol adapter and SUT. The protocol adapter calls ActiveSyncClient to connect, disconnect, and execute commands. The ActiveSyncClient passes the protocol adapter requests and sends the protocol adapter request to the SUT. The ActiveSyncClient receives the response from the SUT and sends the response back to the protocol adapter.

### Helper methods

The common library defines a series of helper methods. The helper methods can be classified into following categories.

* Access the properties in the configuration file.
* Generate resource name.
* Other methods which are used by multiple test suites.

### Message structures

Becuase the C# proxy class is used by the multiple test suites. So the the C# proxy class is defined in the common library.

## Adapter

Adapters are interfaces between the test suites and the SUT. There are two types of adapter: protocol adapter and SUT control adapter. In most cases, modifications to the protocol adapter will not be required for non-Microsoft SUT implementations. However, the SUT control adapter should be appropriately configured to connect to a non-Microsoft SUT implementation. All test suites in the package contain a protocol adapter, six of them contain a SUT control adapter.

### Protocol Adapter

The protocol adapter is a managed adapter, which is derived from the ManagedAdapterBase class in the PTF. It provides an interface that is used by the test cases to construct protocol request messages that will be sent to the SUT. The protocol adapter also acts as an intermediary between the test cases and the transport classes, receiving messages, sending messages, parsing responses from the transport classes, and validating the SUT response according to the normative requirement in the technical specification.

All protocol adapters use ActiveSyncClient transport classes defined in the common library to send and receive messages.

### SUT Control Adapter

The SUT control adapter manages all the control functions of the test suites that are not associated with the protocol. For example, the setup and tear down are managed through the SUT control adapter. The SUT control adapter is designed to work with the Microsoft implementation of the SUT. However, it is configurable to allow the test suites to run against non-Microsoft implementations of the SUT.

There are four protocols that have a SUT control adapter in the Exchange EAS Protocol test suites package: MS-ASCMD, MS-ASHTTP, MS-ASPROV and MS-ASRM.

## Test suite

The test suites verify the server-side and testable requirements listed in the requirement specification. The test suites call the protocol adapter to send and receive message between the protocol adapter and the SUT, and call the SUT control adapter to change the SUT state. The test suites consists of a series test cases which are categorized to several scenarios.

### MS-ASAIRS

Six scenarios are designed to verify the server-side, testable requirements in MS-ASAIRS test suite. The following table lists the scenarios designed in the test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_BodyPartPreference | Test the BodyPartPreference element and BodyPart element in the AirSyncBase namespace, which is used by the Sync command, Search command and ItemOperations command to identify the data sent by and returned to client. |
| S02\_BodyPreference | Test the BodyPreference element and Body element in the AirSyncBase namespace, which is used by the Sync command, Search command and ItemOperations command to identify the data sent by and returned to client. |
| S03\_Attachment | Test the Attachments element and its subelements in the AirSyncBase namespace, which is used by the Sync command, Search command and ItemOperations command to identify the data sent by and returned to client. |
| S04\_StatusError | Test the status error which is returned by the Sync command, Search command and ItemOperations command when the XML elements in AirSyncBase namespace don't comply with the requirements regarding data type, number of instance, order and placement in the XML hierarchy. |
| S05\_Location | This scenario is designed to test the Location element and its sub elements, which is used by the Sync command, Search command and ItemOperations command. |
| S06\_MeetingResponseCommand | This scenario is designed to test the MeetingResponse command. |

### MS-ASCAL

Two scenarios are designed to verify the server-side, testable requirements in MS-ASCAL test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CalendarElement | This scenario is to test Calendar class elements, which are not attached in a Meeting request, including synchronizing the calendar on the server, fetching information of the calendar or searching a specific calendar. |
| S02\_MeetingElement | This scenario is to test Calendar class elements, which are attached in a Meeting request, when meeting is either accepted, tentative accepted, cancelled or declined. |

### MS-ASCMD

22 scenarios are designed to verify the server-side, testable requirements in MS-ASCMD test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_Autodiscover | Discover the account settings by using the Autodiscover command. |
| S02\_FolderCreate | Create the folders by using the FolderCreate command. |
| S03\_FolderDelete | Delete the folders by using the FolderDelete command. |
| S04\_FolderSync | Synchronize the folders by using the FolderSync command. |
| S05\_FolderUpdate | Update the folders by using the FolderUpdate command. |
| S06\_GetAttachment | Retrieve an email attachment from the server by using GetAttachment command. |
| S07\_GetItemEstimate | Get an estimate of the number of items in a collection or folder on the server that have to be synchronized by using GetItemEstimate command. |
| S08\_ItemOperations | Provide the batched online handling of Fetch, empty the folder contents and move the operations against the server by using the ItemOperations command. |
| S09\_MeetingResponse | Accept, tentatively accept, or decline a meeting request in the user's Inbox folder or Calendar folder by using the MeetingResponse command. |
| S10\_MoveItems | Move an item or items from one folder on the server to another by using the MoveItems command. |
| S11\_Ping | Ping the server for updating the folder by using the Ping command. |
| S12\_Provision | Download and acknowledge policies of Provision by using the Provision command. |
| S13\_ResolveRecipients | Resolve the recipients and retrieve the free or busy data by using the ResolveRecipients command. |
| S14\_Search | Find the entries in an address book, mailbox, or document library by using the Search command. |
| S15\_SendMail | Send the email message by using the SendMail command. |
| S16\_Settings | Set or change the device information, OOF setting and user information by using the Settings command. |
| S17\_SmartForward | Forward the email message by using the SmartForward command. |
| S18\_SmartReply | Reply the email message by using the SmartReply command. |
| S19\_Sync | Synchronize the data by using the Sync command. |
| S20\_ValidateCert | Validate a certificate that has been received via an S/MIME mail by using ValidateCert command. |
| S21\_CommonStatusCode | Test the common negative status for the commands. |
| S22\_GetHierarchy | This scenario is designed to test the GetHierarchy command. |

### MS-ASCNTC

Three scenarios are designed to verify the server-side, testable requirements in MS-ASCNTC test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_Sync | This scenario is designed to use the Sync command to synchronize the Contact class data between client and server. |
| S02\_ItemOperations | This scenario is designed to use ItemOperations command to retrieve Contact class data from the server. |
| S03\_Search | This scenario is designed to use Search command to search Contact class data on the server. |

### MS-ASCON

Five scenarios are designed to verify the server-side, testable requirements in MS-ASCON test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_Sync | This scenario is designed to mark a conversation as Read or Unread, flag a conversation for follow-up, apply a conversation-based filter, delete a conversation and request a Message part using Sync command. |
| S02\_GetItemEstimate | This scenario is designed to apply a conversation-based filter using GetItemEstimate command. |
| S03\_ItemOperations | This scenario is designed to ignore a conversation, set up a conversation to be moved always and request a Message part using ItemOperations command. |
| S04\_MoveItems | This scenario is designed to move a conversation from the current folder using MoveItems command. |
| S05\_Search | This scenario is designed to find a conversation using Search command. |

### MS-ASDOC

Two scenarios are designed to verify the server-side, testable requirements in MS-ASDOC test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SearchCommand | Retrieve Document class items that match the criteria specified by the client through the Search command messages. |
| S02\_ItemOperationsCommand | Retrieve data from the server for one or more individual documents through ItemOperations command messages. |

### MS-ASEMAIL

Four scenarios are designed to verify the server-side, testable requirements in MS-ASEMAIL test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_Email | Test normal e-mail events, including sending an e-mail to server, synchronizing e-mails with server, retrieving e-mail items that match the criteria specified by the client from server, retrieving data from the server for one or more specific e-mail items. |
| S02\_EmailVoiceAttachment | Test voice attachment e-mail events, including sending an e-mail with voice attachment to server, synchronizing the e-mail with voice attachment with server. |
| S03\_EmailFlag | Test flag events, include setting a flag on email or task, updating a flag on email or task, marking a flag on email or task as complete, clearing a flag from email or task. |
| S04\_MeetingRequest | Test meeting request events, including sending a meeting request to server, synchronizing the meeting request with server. |

### MS-ASHTTP

Four scenarios are designed to verify the server-side, testable requirements in MS-ASHTTP test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_HTTPPOSTPositive | Test the positive behaviors issued by HTTP POST command. |
| S02\_HTTPPOSTNegative | Test the negative behaviors issued by HTTP POST command. |
| S03\_HTTPPOSTOptionalHeader | Test optional request header of HTTP POST command. |
| S04\_HTTPOPTIONSMessage | Test HTTP OPTIONS command. |

### MS-ASNOTE

Three scenarios are designed to verify the server-side, testable requirements in MS-ASNOTE test suite. The following table lists the scenarios designed in this test suite

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SyncCommand | Synchronize Notes class items for a specified user with the existing notes stored on the server. |
| S02\_SearchCommand | Retrieve Notes class items that match the criteria specified by the client. |
| S03\_ItemOperationsCommand | Retrieve data from the server for one or more notes items. |

### MS-ASPROV

Three scenarios are designed to verify the server-side, testable requirements in MS-ASPROV test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_AcknowledgePolicySettings | Test the acknowledgement phase of Provision. |
| S02\_RemoteWipe | Test the remote wipe directive. |
| S03\_ProvisionNegative | Test negative status of Provision command. |

### MS-ASRM

Five scenarios are designed to verify the server-side and testable requirements in MS-ASRM test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_Settings\_SendMail\_Sync | This scenario is designed to get templateIDs, send email messages and synchronize data from the server by using Settings, SendMail and Sync commands. |
| S02\_ItemOperations | This scenario is designed to fetch a rights-managed e-mail message with or without RemoveRightsManagementProtection element by using ItemOperations command. |
| S03\_Search | This scenario is designed to find rights-managed e-mail messages by using Search command. |
| S04\_SmartForward | This scenario is designed to forward messages by using SmartForward command. |
| S05\_SmartReply | This scenario is designed to reply messages by using SmartReply command. |

### MS-ASTASK

Three scenarios are designed to verify the server-side, testable requirements in MS-ASTASK test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SyncCommand | This scenario is to test Task class element on the server by using Sync command. |
| S02\_ItemOperationsCommand | This scenario is to test Task class element on the server by using ItemOperations command. |
| S03\_SearchCommand | This scenario is to test Task class element on the server by using Search command. |