

MS-ASCNTC Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how MS-ASCNTC test suite is designed to test MS-ASCNTC Open Specification usability and accuracy. It describes test assumptions, scope and constraints of the test suite. It also specifies test scenarios, detailed test cases, test suite architecture and adapter design.

Contents

[1 Configuring the test suite 3](#_Toc387411505)

[1.1 Configuring the test suite client 3](#_Toc387411506)

[1.1.1 Configuring the test suite client manually 3](#_Toc387411507)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc387411508)

[1.2 Configuring the system under test (SUT) 3](#_Toc387411509)

[1.2.1 Configuring the SUT manually 3](#_Toc387411510)

[1.2.2 Configuring the SUT by scripts 3](#_Toc387411511)

[1.3 Configuring the SHOULD/MAY requirements 3](#_Toc387411512)

[2 Test suite design 5](#_Toc387411513)

[2.1 Assumptions, scope and constraints 5](#_Toc387411514)

[2.2 Test suite architecture 5](#_Toc387411515)

[2.3 Technical dependencies and considerations 7](#_Toc387411516)

[2.4 Adapter design 7](#_Toc387411517)

[2.4.1 Adapter overview 7](#_Toc387411518)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387411519)

[2.4.3 Adapter abstract layer 8](#_Toc387411520)

[2.4.4 Adapter details 8](#_Toc387411521)

[2.5 Test scenarios 9](#_Toc387411522)

[2.5.1 S01\_Sync 10](#_Toc387411523)

[2.5.2 S02\_ItemOperations 10](#_Toc387411524)

[2.5.3 S03\_Search 10](#_Toc387411525)

[2.6 Test case design 12](#_Toc387411526)

[2.6.1 Traditional test case design 12](#_Toc387411527)

[2.6.2 Test case description 12](#_Toc387411528)

# Configuring the test suite

## Configuring the test suite client

### Configuring the test suite client manually

Before you run the test suite, update the values in the MS-ASCNTC\_TestSuite.deployment.ptfconfig file. The MS-ASCNTC\_TestSuite.deployment.ptfconfig file can also be configured by running the client setup script.

1. Open MS-ASCNTC\TestSuite\MS-ASCNTC\_TestSuite.deployment.ptfconfig.
2. Update the following value to specify the common configuration file.

Property name="CommonConfigurationFileName" value="ExchangeCommonConfiguration.deployment.ptfconfig"

**Note** This property can be removed or set to empty if the required properties are copied to the test suite specific configuration file. Any other changes to this property will cause all test cases in the test suite to fail during execution. The test suite searches through its specific configuration file and uses those properties, if they are defined, before looking for them in the common configuration file (if specified).

1. Update the following properties' values to match SUT settings and configuration.

* Property name="User1Name" value="MSASCNTC\_User01"
* Property name="User1Password" value="Password01!"
* Property name="User2Name" value="MSASCNTC\_User02"
* Property name="User2Password" value="Password01!"

### Configuring the test suite client by scripts

To configure the test suite client using scripts, see section 5.2.1 of the [ExchangeEASTestSuiteDeploymentGuide.docx](../ExchangeEASTestSuiteDeploymentGuide.docx).

## Configuring the system under test (SUT)

### Configuring the SUT manually

To manually configure the SUT, see section 5.1.2 of the [ExchangeEASTestSuiteDeploymentGuide.docx](../ExchangeEASTestSuiteDeploymentGuide.docx).

### Configuring the SUT by scripts

To configure the SUT using scripts, see section 5.1.1 of the [ExchangeEASTestSuiteDeploymentGuide.docx](../ExchangeEASTestSuiteDeploymentGuide.docx).

## Configuring the SHOULD/MAY requirements

Implementation of the SHOULD/MAY and endnote-related requirements are pre-configured in the format "<Property name="RXXXEnabled" value= "XXXX"/>" for the product versions in the following config files:

* MS-ASCNTC\_ExchangeServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-ASCNTC\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-ASCNTC\_ExchangeServer2013\_SHOULDMAY.deployment.ptfconfig

If RXXXEnabled is set to true, the requirement must be checked. If false, the requirement must not be checked. For Microsoft product versions, all values should not be changed. For third-party products, the closest Microsoft product version should be chosen, and the value of RXXXEnabled should be updated according to the real product behavior. For example, if Exchange Server 2010 is chosen,user can open **MS-ASCNTC\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig** and update the RXXXEnabled accordingly.

# Test suite design

## Assumptions, scope and constraints

Assumptions

None.

Scope

In scope

* This test suite will verify the accuracy and integrity of the technical content in the Open Specification against the results returned from the protocol server by using five commands: FolderSync, SendMail, Sync, Search and ItemOperations.
* This test suite will verify all the XML schema elements embedded inside of the command response.
* This test suite will verify the server-side and testable requirements by running all the test cases on both HTTP and HTTPS.
* This test suite will verify requirements from Exchange ActiveSync: Data Types ([MS-ASDTYPE]) and Exchange ActiveSync: WAP Binary XML Algorithm ([MS-ASWBXML]).

Out of scope

* This test suite will not verify the requirements related to client behaviors.
* This test suite will not verify the requirements related to server internal behaviors.
* This test suite will not verify the internal implementations of its transport protocol stack.

Constraints

None.

## Test suite architecture

This test suite verifies the server-side and testable requirements obtained from the Open Specification. The following figure shows the architecture of this test suite.



The architecture of the test suite

The details of the MS-ASCNTC test suite architecture

* SUT hosts the ActiveSync service which this test suite runs against.
* From third-party user’s point of view, the SUT is the protocol server implementation.
* The following products have been tested with the MS-ASCNTC test suite 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)
* The test suite acts as the client to communicate with the SUT and validates the requirements gathered from the MS-ASCNTC Open Specification.
* Test cases use the MS-ASCNTC adapter to call and get the results of the MS-ASCNTC commands.
* MS-ASCNTC adapter is used in the test cases. The test cases call the methods in the interfaces to invoke the MS-ASCNTC protocol adapter’s commands.
* MS-ASCNTC adapter uses ActiveSyncClient to send command request and retrieve command response.
* ActiveSyncClient encodes and decodes FolderSync command, SendMail command, Sync command, Search command and ItemOperations command defined in [MS-ASCMD] by using MS-ASWBXML and communicates with the SUT via MS-ASHTTP.

## Technical dependencies and considerations

Dependencies

* This test suite depends on the HTTP protocol or HTTPS protocol to transmit the messages.
* This test suite depends on the xsd.exe tool in the .NET Framework SDK to generate structures used in the MS-ASCNTC request and response.
* This test suite depends on the Protocol Test Framework (PTF) to derive managed adapter.
* This test suite depends on MS-ASWBXML to encode XMLrequest bodies into Wireless Application Protocol (WAP) Binary XML (WBXML) for transmission to an ActiveSync server.
* This test suite depends on MS-ASHTTP to synchronize data that is stored on the server.

Encryption consideration

* Transportation of MS-ASCNTC includes HTTP and HTTPS, and encryption will be handled by HTTPS.

## Adapter design

### Adapter overview

One protocol adapter is used in this test suite.

Protocol adapter

* MS-ASCNTC protocol adapter
* The MS-ASCNTC adapter is a managed adapter, which is derived from the ManagedAdapterBase class in PTF.
* The MS-ASCNTC adapter has the following functionalities
* Choose HTTP or HTTPS for transport;
* Construct requests of the five commands used in MS-ASCNTC;
* Communicate with the SUT by sending requests to the SUT and receiving the corresponding responses from the SUT;
* Parse the response messages and validate the messages according to the XML schema.

### Technical feasibility of adapter approach

Message generation

The MS-ASCNTC adapter gets the parameter values and calls the corresponding commands in ActiveSyncClient. The ActiveSyncClient serializes the parameter values to XML elements, and then encodes the request bodies into WBXML for transmission to an ActiveSync server.

Message consumption

The messages received from the SUT will be parsed in the ActiveSyncClient and be passed to the MS-ASCNTC adapter. The messages are then consumed in the MS-ASCNTC adapter to validate the message format and to validate the logic-related requirements in the test cases.

### Adapter abstract layer

Protocol adapter

MS-ASCNTC adapter interface

There are six methods declared in the MS-ASCNTC adapter interface IMS\_ASCNTCAdapter.

Three methods ItemOperations, Search and Sync correspond to the three MS-ASCNTC commands: ItemOperations, Search and Sync. And the other three methods FolderSync, SendMail and SwitchUser are used to synchronize the collection hierarchy, send MIME-formatted e-mail messages and switch current user to another to call ActiveSync commands.

The methods are described in the following table.

|  |  |  |
| --- | --- | --- |
| **No.** | **Methods** | **Description** |
| 1 | FolderSync | Synchronizes the collection hierarchy. |
| 2 | ItemOperations | Retrieves an item from the server. |
| 3 | Search | Finds entries in an address book, mailbox or document library. |
| 4 | SendMail | Sends MIME-formatted e-mail messages to the server. |
| 5 | SwitchUser | Changes user to call ActiveSync command. |
| 6 | Sync | Synchronizes the changes in a collection between the client and the server. |

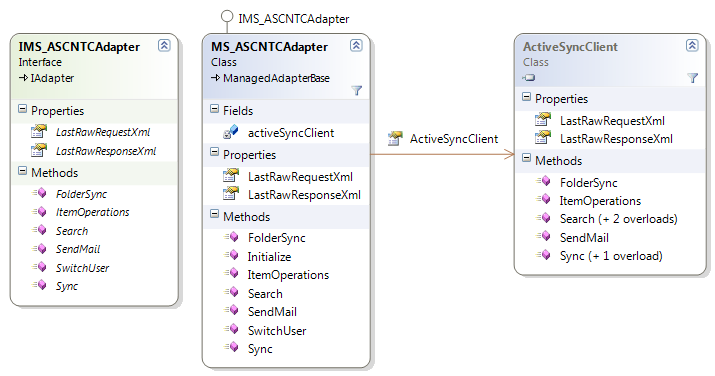
MS-ASCNTC adapter interface methods

### Adapter details

#### Protocol adapter

##### MS-ASCNTC protocol adapter

The following figure shows the class diagram of the MS-ASCNTC protocol adapter and the relationship between MS\_ASCNTCAdapter and ActiveSyncClient.



MS-ASCNTC adapter and ActiveSyncClient class diagram

The following outlines details of the class diagram:

Adapter interface

* IMS\_ASCNTCAdapter is the interface of the protocol adapter.
* IMS\_ASCNTCAdapter defines the methods invoked by test cases, including FolderSync, ItemOperations, Search, SendMail, SwitchUser and Sync methods.

Adapter implementation

* MS\_ASCNTCAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_ASCNTCAdapter.
* MS\_ASCNTCAdapter invokes the methods defined in ActiveSyncClient to synchronize information, find entries, retrieve items, send e-mail messages and switch user.
* The Initialize method is used to initialize an instance of ActiveSyncClient.
* The two properties LastRawRequestXml and LastRawResponseXml are used to get the raw request and response xml data.

## Test scenarios

Three scenarios are designed to cover the server-side, testable requirements in the MS-ASCNTC test suite. The details of the scenarios are as follows.

|  |  |
| --- | --- |
| 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-ASCNTC scenarios

### S01\_Sync

Description

This scenario is designed to use the Sync command to synchronize the Contact class data between client and server.

Commands

* Sync
* SendMail
* FolderSync

Prerequisites

Call FolderSync command to get ServerId of Inbox, Contacts and Recipient Information Cache folders.

Cleanup

Call Sync command to delete the contact items and email messages created in test cases.

### S02\_ItemOperations

Description

This scenario is designed to use ItemOperations command to retrieve Contact class data from the server.

Commands

* Sync
* ItemOperations
* FolderSync

Prerequisites

Call FolderSync command to get ServerId of Inbox, Contacts and Recipient Information Cache folders.

Cleanup

Call Sync command to delete the contact items created in test cases.

### S03\_Search

Description

This scenario is designed to use Search command to search Contact class data on the server.

Commands

* Sync
* Search
* FolderSync

Prerequisites

Call FolderSync command to get ServerId of Inbox, Contacts and Recipient Information Cache folders.

Cleanup

Call Sync command to delete the contact items created in test cases.

## Test case design

### Traditional test case design

Traditional testing is adopted as the test approach for this test suite. The test cases are designed to cover the server-side and testable requirements.

There are 13 traditional test cases designed to cover the three scenarios mentioned in section [2.6 Test scenarios](#_Test_scenarios). Details of the traditional test cases are specified in section [2.7.2 Test case description](#_Test_case_description). The scenarios distributions of the test cases are listed in the following table.

|  |  |
| --- | --- |
| Scenario ID | Test case name |
| S01\_Sync | MSASCNTC\_S01\_TC01\_Sync\_AddContact |
| MSASCNTC\_S01\_TC02\_Sync\_GhostedElements\_ExceptAssistantName |
| MSASCNTC\_S01\_TC03\_Sync\_GhostedElement\_AssistantName |
| MSASCNTC\_S01\_TC04\_Sync\_RefreshRecipientInformationCache |
| MSASCNTC\_S01\_TC05\_Sync\_Status6\_PictureExceeds48KB |
| MSASCNTC\_S01\_TC06\_Sync\_AddContact\_300CategoryElements |
| MSASCNTC\_S01\_TC07\_Sync\_AddContact\_300ChildElements |
| MSASCNTC\_S01\_TC08\_Sync\_TruncatedBody |
| MSASCNTC\_S01\_TC09\_Sync\_NonTruncatedBody |
| S02\_ItemOperations | MSASCNTC\_S02\_TC01\_ItemOperations\_RetrieveContact |
| MSASCNTC\_S02\_TC02\_ItemOperations\_TruncateBody |
| MSASCNTC\_S02\_TC03\_ItemOperations\_SchemaViewFetch |
| S03\_Search | MSASCNTC\_S03\_TC01\_Search\_RetrieveContact |

Test case scenario distribution

### Test case description

The following table describes the common prerequisites and common cleanup for all the test cases.

|  |  |
| --- | --- |
| Common prerequisites | 1. Use HTTP or HTTPS as the transport. 2. Call FolderSync command to get ServerId of Inbox, Contacts and Recipient Information Cache folders. |
| Common cleanup | Call Sync command to delete the contact items and email messages created in test cases. |

Test case common prerequisites and common cleanup

The steps in the following test cases use methods and parameters in the adapter interfaces directly.

The following tables describe the traditional test cases.

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC01\_Sync\_AddContact |
| Description | This case is designed to use Sync Add operation to add a contact. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact with all Contact class elements to the server. 2. The client calls Sync command to synchronize the contact item that was added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC01\_Sync\_AddContact

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC02\_Sync\_GhostedElements\_ExceptAssistantName |
| Description | This case is designed to test ghosted elements except AssistantName element. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls Sync command with Supported element in initial Sync request to synchronize the contact item that was added in previous step. 3. The client calls Sync command with Change element to change the AssistantName value of the contact. 4. The client calls the Sync command to synchronize the changed contact on the server. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC02\_Sync\_GhostedElements\_ExceptAssistantName

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC03\_Sync\_GhostedElement\_AssistantName |
| Description | This case is designed to test the AssistantName element which can be ghosted. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls Sync command with Supported element in initial Sync request to synchronize the contact item that was added in previous step. 3. The client calls Sync command with Change element to change the value of one element, such as JobTitle element of the contact. 4. The client calls Sync command with Supported element in initial Sync request to synchronize the changed contact on the server. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC03\_Sync\_GhostedElement\_AssistantName

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC04\_Sync\_RefreshRecipientInformationCache |
| Description | This case is designed to retrieve a minimal set of Contact class data from the server by issuing a Sync command request against the recipient information cache. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls SendMail command to send a mail to recipient. 2. The client calls Sync command to synchronize the recipient information cache folder of the sender. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC04\_Sync\_RefreshRecipientInformationCache

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC05\_Sync\_Status6\_PictureExceeds48KB |
| Description | This case is designed to test server must return a status error of 6 if the value of the Picture element exceeds 48 KB. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact with a picture whose value exceeds 48 KB. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC05\_Sync\_Status6\_PictureExceeds48KB

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC06\_Sync\_AddContact\_300CategoryElements |
| Description | This case is designed to test the Categories element can have up to 300 Category elements. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with 300 Category elements per Categories element to add a contact to the server. 2. The client calls Sync command to synchronize the contact added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC06\_Sync\_AddContact\_300CategoryElements

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC07\_Sync\_AddContact\_300ChildElements |
| Description | This case is designed to test the Children element can have up to 300 Child elements. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with 300 Child elements per Children element to add a contact to the server. 2. The client calls Sync command to synchronize the contact added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC07\_Sync\_AddContact\_300ChildElements

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC08\_Sync\_TruncatedBody |
| Description | This case is designed to test server truncates the contents of the airsyncbase:Body element in the Sync command response if the client requests truncation. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls Sync command with TruncationSize element smaller than the available data the body to synchronize the contact. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC08\_Sync\_TruncatedBody

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCNTC\_S01\_TC09\_Sync\_NonTruncatedBody |
| Description | This case is designed to test server will no longer truncate the contents of the airsyncbase:Body element if the client sends an airsyncbase:BodyPreference element in the request that contains a Type element to specify the desired format, but does not include the airsyncbase:TruncationSize element. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls Sync command without TruncationSize element to synchronize the contact. |
| Cleanup | Common cleanup |

MSASCNTC\_S01\_TC09\_Sync\_NonTruncatedBody

|  |  |
| --- | --- |
| S02\_ItemOperations | |
| Test case ID | MSASCNTC\_S02\_TC01\_ItemOperations\_RetrieveContact |
| Description | This case is designed to retrieve contact item using ItemOperations command. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls ItemOperations command to retrieve the contact item that added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S02\_TC01\_ItemOperations\_RetrieveContact

|  |  |
| --- | --- |
| S02\_ItemOperations | |
| Test case ID | MSASCNTC\_S02\_TC02\_ItemOperations\_TruncateBody |
| Description | This case is designed to test server truncates the contents of the airsyncbase:Body element in the ItemOperations command response if the client requests truncation. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls ItemOperations command with TruncationSize element smaller than the available data the body. |
| Cleanup | Common cleanup |

MSASCNTC\_S02\_TC02\_ItemOperations\_TruncateBody

|  |  |
| --- | --- |
| S02\_ItemOperations | |
| Test case ID | MSASCNTC\_S02\_TC03\_ItemOperations\_SchemaViewFetch |
| Description | This case is designed to test if an airsync:Schema element is included in the ItemOperations command request; server response must be restricted to the elements that were included as child elements of the airsync:Schema element in the command request. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls ItemOperations command with Schema element to retrieve the contact item that added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S02\_TC03\_ItemOperations\_SchemaViewFetch

|  |  |
| --- | --- |
| S03\_Search | |
| Test case ID | MSASCNTC\_S03\_TC01\_Search\_RetrieveContact |
| Description | This case is designed to retrieve contact using Search command. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls Sync command with Add element to add a contact to the server. 2. The client calls Search command to retrieve the contact item that added in previous step. |
| Cleanup | Common cleanup |

MSASCNTC\_S03\_TC01\_Search\_RetrieveContact