

MS-ASCON Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how the MS-ASCON test suite is designed to test the MS-ASCON 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](#_Toc387411602)

[1.1 Configuring the test suite client 3](#_Toc387411603)

[1.1.1 Configuring the test suite client manually 3](#_Toc387411604)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc387411605)

[1.2 Configuring the system under test (SUT) 3](#_Toc387411606)

[1.2.1 Configuring the SUT manually 3](#_Toc387411607)

[1.2.2 Configuring the SUT by scripts 3](#_Toc387411608)

[1.3 Configuring the SHOULD/MAY requirements 3](#_Toc387411609)

[2 Test suite design 5](#_Toc387411610)

[2.1 Assumptions, scope and constraints 5](#_Toc387411611)

[2.2 Test suite architecture 5](#_Toc387411612)

[2.3 Technical dependencies and considerations 7](#_Toc387411613)

[2.4 Adapter design 7](#_Toc387411614)

[2.4.1 Adapter overview 7](#_Toc387411615)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387411616)

[2.4.3 Adapter abstract layer 8](#_Toc387411617)

[2.4.4 Adapter details 8](#_Toc387411618)

[2.5 Test scenarios 9](#_Toc387411619)

[2.5.1 S01\_Sync 10](#_Toc387411620)

[2.5.2 S02\_GetItemEstimate 10](#_Toc387411621)

[2.5.3 S03\_ItemOperations 11](#_Toc387411622)

[2.5.4 S04\_MoveItems 11](#_Toc387411623)

[2.5.5 S05\_Search 11](#_Toc387411624)

[2.6 Test case design 12](#_Toc387411625)

[2.6.1 Traditional test case design 12](#_Toc387411626)

[2.6.2 Test case description 13](#_Toc387411627)

# 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-ASCON\_TestSuite.deployment.ptfconfig file. The MS-ASCON\_TestSuite.deployment.ptfconfig file can also be configured by running the client setup script.

1. Open MS-ASCON\TestSuite\MS-ASCON\_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="MSASCON\_User01"
* Property name="User1Password" value="Password01!"
* Property name="User2Name" value="MSASCON\_User02"
* Property name="User2Password" value="Password01!"
* Property name="User3Name" value="MSASCON\_User03"
* Property name="User3Password" 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-ASCON\_ExchangeServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-ASCON\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-ASCON\_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-ASCON\_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 nine commands: Sync, Search, ItemOperations, FolderSync, SendMail, SmartReply, SmartForward, MoveItems and GetItemEstimate.
* 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 test cases on both HTTP and HTTPS.
* This test suite will verify requirements from Exchange ActiveSync: WAP Binary XML (WBXML) Algorithm ([MS-ASWBXML]) and Exchange ActiveSync: Data Types ([MS-ASDTYPE]).

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-ASCON 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-ASCON test suite on the Windows platform.
* 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-ASCON Open Specification.
* Test cases use the MS-ASCON protocol adapter to call and get the results of the ActiveSync commands.
* MS-ASCON protocol adapter is used in the test cases. The test cases call the methods in the interfaces to invoke the MS-ASCON protocol adapter’s commands.
* MS-ASCON protocol adapter uses ActiveSyncClient to send command request and retrieve command response.
* ActiveSyncClient encodes and decodes Sync, Search, ItemOperations, FolderSync, SendMail, SmartReply, SmartForward, MoveItems and GetItemEstimate 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 HTTP protocol or HTTPS protocol to transmit the messages.
* This test suite depends on Protocol Test Framework (PTF) to derive managed adapters.
* This test suite depends on MS-ASWBXML to encode request bodies into WBXML for transmission to an ActiveSync server.
* This test suite depends on the xsd.exe tool in the .NET Framework SDK to generate structures used in the MS-ASCON request and response.
* This test suite depends on MS-ASHTTP to synchronize data with the data that is stored on the server.

Encryption consideration

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

### Technical feasibility of adapter approach

Message generation

The MS-ASCON adapter gets the parameter values and calls the corresponding commands in ActiveSyncClient. The ActiveSyncClient serializes the parameter values to XML elements, and then encodes 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 upon to the MS-ASCON adapter. Then these messages are consumed in the MS-ASCON adapter to validate the message format and to validate the logic-related requirements in the test cases.

### Adapter abstract layer

Protocol adapter

MS-ASCON adapter interface

There are 11 methods declared in the MS-ASCON adapter interface IMS\_ASCONAdapter.

Five methods Sync, Search, ItemOperations, MoveItems and GetItemEstiamte correspond to the five MS-ASCON commands: Sync, Search, ItemOperations, MoveItems and GetItemEstimate. And the other six methods SyncEmail, FolderSync, SendMail, SmartReply, SmartForward and SwitchUser are used to synchronize email with specific subject, synchronize the collection hierarchy, send, reply and forward email, and switch current user to another to call ActiveSync commands.

The methods are described in the following table.

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1 | Sync | This method is used to synchronize changes in a collection between the client and the server. |
| 2 | Search | This method is used to find entries address book, mailbox, or document library. |
| 3 | ItemOperations | This method is used to act as a container for the Fetch element, the EmptyFolderContents element, and the Move element to provide batched online handling of these operations against the server. |
| 4 | FolderSync | This method is used to synchronize the collection hierarchy. |
| 5 | SendMail | This method is used to send MIME-formatted e-mail messages to the server. |
| 6 | SmartReply | This method is used to reply to messages without retrieving the full, original message from the server. |
| 7 | SmartForward | This method is used to forward messages without retrieving the full, original message from the server. |
| 8 | MoveItems | This method is used to move an item or items from one folder on the server to another. |
| 9 | GetItemEstimate | This method is used to get an estimate of the number of items in a collection or folder on the server that have to be synchronized. |
| 10 | SwitchUser | This method is used to change the user authentication. |
| 11 | SyncEmail | This method is used to find an email with specific subject. |

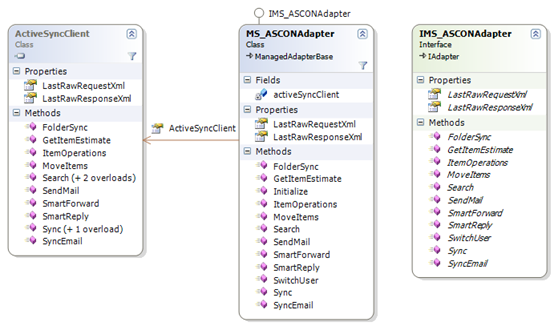
MS-ASCON adapter interface methods

### Adapter details

#### Protocol adapter

##### MS-ASCON protocol adapter

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



MS-ASCON adapter and ActiveSyncClient class diagram

The following outlines details of the class diagram:

Adapter interface

* IMS\_ASCONAdapter is the interface of the protocol adapter.
* IMS\_ASCONAdapter defines the methods invoked by test cases, including Sync, Search, ItemOperations, FolderSync, SendMail, SmartReply, SmartForward, MoveItems, GetItemEstimate, SwitchUser and SyncEmail methods.

Adapter implementation

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

## Test scenarios

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

|  |  |
| --- | --- |
| 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-ASCON scenarios

### S01\_Sync

Description

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.

Commands

* Sync
* SmartReply
* SendMail
* SmartForward

Prerequisites

* Calls SwitchUser to switch the current user to User1 and calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder.

Cleanup

* Calls Sync command to delete the items created in test cases.

### S02\_GetItemEstimate

Description

This scenario is designed to apply a conversation-based filter using GetItemEstimate command.

Commands

* Sync
* SendMail
* SmartReply
* GetItemEstimate

Prerequisites

* Calls SwitchUser to switch the current user to User1 and calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder.

Cleanup

* Calls Sync command to delete the items created in test cases.

### S03\_ItemOperations

Description

This scenario is designed to ignore a conversation, set up a conversation to be moved always and request a Message part using ItemOperations command.

Commands

* Sync
* SendMail
* ItemOperations
* SmartReply

Prerequisites

* Calls SwitchUser to switch the current user to User1 and calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder.

Cleanup

* Calls Sync command to delete the items created in test cases.

### S04\_MoveItems

Description

This scenario is designed to move a conversation from the current folder using MoveItems command.

Commands

* Sync
* SendMail
* SmartReply
* MoveItems

Prerequisites

* Calls SwitchUser to switch the current user to User1 and calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder.

Cleanup

* Calls Sync command to delete the items created in test cases.

### S05\_Search

Description

This scenario is designed to find a conversation using Search command.

Commands

* Sync
* SendMail
* SmartReply
* Search

Prerequisites

* Calls SwitchUser to switch the current user to User1 and calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder.

Cleanup

* Calls Sync command to delete the 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 22 traditional test cases are designed to cover the five scenarios mentioned in section [2.6 Test scenarios](#_Test_scenarios_1). Details of traditional test cases are specified in [section 2.7.2 Test case description](#_Test_Cases_Description). The scenarios distributions of the test cases are listed in the following table.

|  |  |
| --- | --- |
| Scenario ID | Test case name |
| S01\_Sync | MSASCON\_S01\_TC01\_Sync\_MarkRead |
| MSASCON\_S01\_TC02\_Sync\_Flag |
| MSASCON\_S01\_TC03\_Sync\_Delete |
| MSASCON\_S01\_TC04\_Sync\_Filter |
| MSASCON\_S01\_TC05\_Sync\_ConversationIndex |
| MSASCON\_S01\_TC06\_Sync\_NoConversationId |
| MSASCON\_S01\_TC07\_Sync\_MessagePart |
| MSASCON\_S01\_TC08\_Sync\_Status164 |
| S02\_GetItemEstimate | MSASCON\_S02\_TC01\_GetItemEstimate\_Filter |
| MSASCON\_S02\_TC02\_GetItemEstimate\_Status4 |
| S03\_ItemOperations | MSASCON\_S03\_TC01\_ItemOperations\_Ignore |
| MSASCON\_S03\_TC02\_ItemOperations\_MoveAlways |
| MSASCON\_S03\_TC03\_ItemOperations\_Status2 |
| MSASCON\_S03\_TC04\_ItemOperations\_Status6 |
| MSASCON\_S03\_TC05\_ItemOperations\_Status105 |
| MSASCON\_S03\_TC06\_ItemOperations\_Status156 |
| MSASCON\_S03\_TC07\_ItemOperations\_Status164 |
| MSASCON\_S03\_TC08\_ItemOperations\_MessagePart |
| S04\_MoveItems | MSASCON\_S04\_TC01\_MoveItems\_Move |
| S05\_Search | MSASCON\_S05\_TC01\_Search |
| MSASCON\_S05\_TC02\_Search\_MessagePart |
| MSASCON\_S05\_TC03\_Search\_Status164 |

Test case scenario distribution

### Test case description

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

|  |  |
| --- | --- |
| Common steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client using User2 calls SendMail command to send mail to User1.  3. The client calls SwitchUser the switch User2 to User1.  4. The client using User1 calls Sync command to get the received email.  5. The client using User1 calls SmartReply command to reply the email to User2.  6. The client calls SwitchUser to switch User1 to User2.  7. The client using User2 calls Sync command to get the received email.  8. The client using User2 calls SmartReply to reply email to User1.  9. The client calls SwitchUser to switch User2 to User1.  10. The client calls Sync command to get the ConversationId and ServerIds of the created conversation. |
| Common Prerequisites | 1. The client calls SwitchUser to switch the current user to User1.  2. The client calls FolderSync command to get ServerId of Inbox, Sent Items, Deleted Items, Calendar and recipient information cache folder. |
| Common cleanup | The client calls Sync command to delete the items created in test cases. |

Test case common steps, 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 | MSASCON\_S01\_TC01\_Sync\_MarkRead |
| Description | This test case is designed to validate marking a conversation as Read or Unread by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls Sync command to mark the created conversation as read.  3. The client calls Sync command to check the Read flags of emails in the conversation.  4. The client calls Sync command to mark the created conversation as unread.  5. The client calls Sync command to check the Read flags of emails in the conversation. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC01\_Sync\_MarkRead

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC02\_Sync\_Flag |
| Description | This test case is designed to validate flagging a conversation for Follow-up by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls Sync command to set flag to the created conversation.  3. The client calls Sync command to check the flag status of the emails in the conversation.  4. The client calls Sync command to clear the flag of the conversation.  5. The client calls Sync command to check the flag status of the emails in the conversation.  6. The client calls Sync command to mark a flagged conversation as complete.  7. The client calls Sync command to check the flag status of the emails in the conversation. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC02\_Sync\_Flag

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC03\_Sync\_Delete |
| Description | This test case is designed to validate deleting a conversation by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls Sync command to delete the created conversation.  3. The client calls Sync command to check Inbox folder and Deleted Items folder.  4. The client calls SwitchUser to switch current user to User2.  5. The client calls Sync command to get the latest email of the created conversation.  6. The client calls SmartReply command to reply the latest email of the created conversation to User1.  7. The client calls SwitchUser to switch User2 to User1.  8. The client calls Sync command to check the received email is not deleted. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC03\_Sync\_Delete

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC04\_Sync\_Filter |
| Description | This test case is designed to validate filtering a conversation by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls Sync command to get the created conversation with setting ConversationMode element to true. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC04\_Sync\_Filter

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC05\_Sync\_ConversationIndex |
| Description | This test case is designed to validate ConversationIndex element by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch User2 to User1.  4. The client calls Sync command to get the received email.  5. The client calls SmartReply command to reply the received email to User2.  6. The client calls SwitchUser to switch User1 to User2.  7. The client calls Sync command to get the received email.  8. The client calls SmartForward command to forward the received email to User3.  9. The client calls SwitchUser to switch User2 to User3.  10. The client calls Sync command to get the received email. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC05\_Sync\_ConversationIndex

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC06\_Sync\_NoConversationId |
| Description | This test case is designed to validate the ConversationId element is not present by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls Sync command to get the initial SyncKey.  2. The client calls Sync command to add a calendar item.  3. The client calls Sync command to get the latest SyncKey.  4. The client calls Sync command to synchronize the Calendar folder. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC06\_Sync\_NoConversationId

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC07\_Sync\_MessagePart |
| Description | This test case is designed to validate requesting the message part by Sync command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch User2 to User1.  4. The client calls Sync command to get the received email without BodyPreference or BodyPartPreference elements.  5. The client calls Sync command to get the received email with only BodyPreference element.  6. The client calls Sync command to get the received email with only BodyPartPrefrernce element.  7. The client calls Sync command to get the received email with both BodyPreference and BodyPartPreference elements. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC07\_Sync\_MessagePart

|  |  |
| --- | --- |
| S01\_Sync | |
| Test case ID | MSASCON\_S01\_TC08\_Sync\_Status164 |
| Description | This test case is designed to validate status 164 is returned if a value other than 2 is specified in the Type element of BodyPartPreference element in Sync command request. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch current user to User1.  4. The client calls Sync command to synchronize the email with setting Type element in the BodyPartPreference element of request to 3. |
| Cleanup | Common cleanup |

MSASCON\_S01\_TC08\_Sync\_Status164

|  |  |
| --- | --- |
| S02\_GetItemEstimate | |
| Test case ID | MSASCON\_S02\_TC01\_GetItemEstimate\_Filter |
| Description | This test case is designed to validate filtering a conversation by GetItemEstimate command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls Sync command to get the latest SyncKey.  3. The client calls GetItemEstimate command to estimate the items in Inbox folder with setting ConversationMode element to true. |
| Cleanup | Common cleanup |

MSASCON\_S02\_TC01\_GetItemEstimate\_Filter

|  |  |
| --- | --- |
| S02\_GetItemEstimate | |
| Test case ID | MSASCON\_S02\_TC02\_GetItemEstimate\_Status4 |
| Description | This test case is designed to validate Status 4 (protocol error) by GetItemEstimate command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls Sync command to synchronize the Calendar folder.  2. The client calls GetItemEstimate command to estimate the items in Calendar folder. |
| Cleanup | Common cleanup |

MSASCON\_S02\_TC02\_GetItemEstimate\_Status4

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC01\_ItemOperations\_Ignore |
| Description | This test case is designed to validate ignoring a conversation by ItemOperations command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command to ignore the conversation.  3. The client calls Sync command to check Inbox folder and Deleted Items folder.  4. The client calls SwitchUser to switch current user to User2.  5. The client calls Sync command to get the latest email of the conversation.  6. The client calls SmartReply command to reply the latest email of the conversation to User1.  7. The client calls SwithUser to switch User2 to User1.  8. The client calls Sync command to check Inbox folder and Deleted Items folder to verify the future email of the conversation will also be moved. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC01\_ItemOperations\_Ignore

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC02\_ItemOperations\_MoveAlways |
| Description | This test case is designed to validate always moving a conversation by ItemOperations command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command to move the conversation from Inbox folder to Sent Items folder with setting MoveAlways element to true.  3. The client calls Sync command to check Inbox folder and Sent Items folder.  4. The client calls SwitchUser to switch current user to User2.  5. The client calls Sync command to get the latest email of the conversation.  6. The client calls SmartReply command to reply the latest email of the conversation to User1.  7. The client calls SwithUser to switch User2 to User1.  8. The client calls Sync command to check Inbox folder and Sent Items folder to verify the future email of the conversation will also be moved. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC02\_ItemOperations\_MoveAlways

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC03\_ItemOperations\_Status2 |
| Description | This test case is designed to validate status 2 is returned for ItemOperations command if the ItemOperations command request is invalid. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command with setting the MoveAlways element to "MoveAlwaysContent". |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC03\_ItemOperations\_Status2

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC04\_ItemOperations\_Status6 |
| Description | This test case is designed to validate status 6 is returned for ItemOperations command if the conversation or destination folder does not exist. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command with an invalid ConversationId.  3. The client calls ItemOperations command with an invalid destination folder id. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC04\_ItemOperations\_Status6

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC05\_ItemOperations\_Status105 |
| Description | This test case is designed to validate status 105 is returned for ItemOperations command if the destination folder is recipient information cache. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command with setting destination folder id to recipient information cache. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC05\_ItemOperations\_Status105

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC06\_ItemOperations\_Status156 |
| Description | This test case is designed to validate status 156 is returned for ItemOperations command if the destination folder is not of type IPF.Note. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls ItemOperations command with setting destination folder id to Calendar folder. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC06\_ItemOperations\_Status156

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC07\_ItemOperations\_Status164 |
| Description | This test case is designed to validate status 164 is returned if a value other than 2 is specified in the Type element of BodyPartPreference element in ItemOperations command request. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch User2 to User1.  4. The client calls ItemOperations command to fetch the email with setting Type element in the BodyPartPreference element of request to 3. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC07\_ItemOperations\_Status164

|  |  |
| --- | --- |
| S03\_ItemOperations | |
| Test case ID | MSASCON\_S03\_TC08\_ItemOperations\_MessagePart |
| Description | This test case is designed to validate requesting the message part by ItemOperations command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch User2 to User1.  4. The client calls ItemOperations command to get the received email without BodyPreference or BodyPartPreference elements.  5. The client calls ItemOperations command to get the received email with only BodyPreference element.  6. The client calls ItemOperations command to get the received email with only BodyPartPrefrernce element.  7. The client calls ItemOperations command to get the received email with both BodyPreference and BodyPartPreference elements. |
| Cleanup | Common cleanup |

MSASCON\_S03\_TC08\_ItemOperations\_MessagePart

|  |  |
| --- | --- |
| S04\_MoveItems | |
| Test case ID | MSASCON\_S04\_TC01\_MoveItems\_Move |
| Description | This test case is designed to validate moving a conversation by MoveItems command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. The client calls MoveItems command to move the Conversation from Inbox folder to Sent Items folder.  3. The client calls Sync command to check the Inbox folder and Sent Items folder to verify the email has been moved. |
| Cleanup | Common cleanup |

MSASCON\_S04\_TC01\_MoveItems\_Move

|  |  |
| --- | --- |
| S05\_Search | |
| Test case ID | MSASCON\_S05\_TC01\_Search |
| Description | This test case is designed to validate searching a conversation by Search command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. Common steps.  2. Call MoveItems command to move one item of the conversation from Inbox folder to Sent Items folder.  3. The client calls Search command to search the specific conversation to check both the items in Inbox folder and Sent Items folder could be found. |
| Cleanup | Common cleanup |

MSASCON\_S05\_TC01\_Search

|  |  |
| --- | --- |
| S05\_Search | |
| Test case ID | MSASCON\_S05\_TC02\_Search\_MessagePart |
| Description | This test case is designed to validate requesting the message part by Search command. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch User2 to User1.  4. The client calls Search command to get the received email without BodyPreference or BodyPartPreference elements.  5. The client calls Search command to get the received email with only BodyPreference element.  6. The client calls Search command to get the received email with only BodyPartPrefrernce element.  7. The client calls Search command to get the received email with both BodyPreference and BodyPartPreference elements. |
| Cleanup | Common cleanup |

MSASCON\_S05\_TC02\_Search\_MessagePart

|  |  |
| --- | --- |
| S05\_Search | |
| Test case ID | MSASCON\_S05\_TC03\_Search\_Status164 |
| Description | This test case is designed to validate Status 164 is returned if a value other than 2 is specified in the Type element of BodyPartPreference element in Search command request. |
| Prerequisites | Common Prerequisites |
| Test execution steps | 1. The client calls SwitchUser to switch the current user to User2.  2. The client calls SendMail command to send email from User2 to User1.  3. The client calls SwitchUser to switch current user to User1.  4. The client calls Search command to search the email with setting Type element in the BodyPartPreference element of request to 3. |
| Cleanup | Common cleanup |

MSASCON\_S05\_TC03\_Search\_Status164