

MS-ASHTTP Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how the MS-ASHTTP test suite is designed to test the MS-ASHTTP 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](#_Toc387411990)

[1.1 Configuring the test suite client 3](#_Toc387411991)

[1.1.1 Configuring the test suite client manually 3](#_Toc387411992)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc387411993)

[1.2 Configuring the system under test (SUT) 3](#_Toc387411994)

[1.2.1 Configuring the SUT manually 3](#_Toc387411995)

[1.2.2 Configuring the SUT by scripts 3](#_Toc387411996)

[1.3 Configuring the SHOULD/MAY requirements 4](#_Toc387411997)

[2 Test suite design 5](#_Toc387411998)

[2.1 Assumptions, scope and constraints 5](#_Toc387411999)

[2.2 Test suite architecture 5](#_Toc387412000)

[2.3 Technical dependencies and considerations 7](#_Toc387412001)

[2.4 Adapter design 7](#_Toc387412002)

[2.4.1 Adapter overview 7](#_Toc387412003)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387412004)

[2.4.3 Adapter abstract layer 8](#_Toc387412005)

[2.4.4 Adapter details 8](#_Toc387412006)

[2.5 Test scenarios 10](#_Toc387412007)

[2.5.1 S01\_HTTPPOSTPositive 10](#_Toc387412008)

[2.5.2 S02\_HTTPPOSTNegative 10](#_Toc387412009)

[2.5.3 S03\_HTTPPOSTOptionalHeader 11](#_Toc387412010)

[2.5.4 S04\_HTTPOPTIONSMessage 11](#_Toc387412011)

[2.6 Test case design 11](#_Toc387412012)

[2.6.1 Traditional test case design 11](#_Toc387412013)

[2.6.2 Test case description 12](#_Toc387412014)

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

1. Open MS-ASHTTP\TestSuite\MS-ASHTTP\_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="MSASHTTP\_User01"
* Property name="User1Password" value="Password01!"
* Property name="User2Name" value="MSASHTTP\_User02"
* Property name="User2Password" value="Password01!"
* Property name="User3Name" value="MSASHTTP\_User03"
* Property name="User3Password" value="Password01!"
* Property name="User4Name" value="MSASHTTP\_User04"
* Property name="User4Password" 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-ASHTTP\_ExchangeServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-ASHTTP\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-ASHTTP\_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-ASHTTP\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig** and update the RXXXEnabled accordingly.

# Test suite design

## Assumptions, scope and constraints

Assumptions

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

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 two commands: HTTP POST and HTTP OPTIONS.
* This test suite will verify the server-side and testable requirements by running all test cases on both HTTP and HTTPS.

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-ASHTTP test suite architecture

* SUT hosts the data synchronization 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-ASHTTP 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 MS-ASHTTP Open Specification.
* Test cases use the MS-ASHTTP protocol adapter to call and get the results of the MS-ASHTTP commands. Test cases also use the SUT control adapter to configure and retrieve information from SUT.
* MS-ASHTTP protocol adapter is used in the test cases. The test cases call the methods in the interfaces to invoke the MS-ASHTTP protocol adapter’s commands.
* MS-ASHTTP protocol adapter 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.
* SUT control adapter is used in the test cases. The test cases call the method in the interface to configure the SUT.

## Technical dependencies and considerations

Technical 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-ASHTTP request and response.

Encryption consideration

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

## Adapter design

### Adapter overview

One protocol adapter and one SUT control adapter are used in this test suite.

Protocol adapter

* MS-ASHTTP protocol adapter
* The MS-ASHTTP adapter is a managed adapter, which is derived from the ManagedAdapterBase class in PTF.
* The MS-ASHTTP adapter has the following functionalities
* Construct requests of two MS-ASHTTP 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 XSD schema;
* Choose HTTP or HTTPS for transport.

SUT control adapters

* SUT control adapter
* The SUT control adapter will be a PowerShell scripted adapter.
* The SUT control adapter has the following functionalities
* Configure the SSL setting in SUT.
* The SUT control adapter is invoked by the test cases.

### Technical feasibility of adapter approach

Message generation

The MS-ASHTTP 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 upon to the MS-ASHTTP adapter. Then these messages are consumed in the MS-ASHTTP adapter to validate the message format and to validate the logic-related requirements in the test cases.

SUT control adapter

The SUT control adapter is designed to remotely control the SUT to configure the SSL setting.

### Adapter abstract layer

Protocol adapter

MS-ASHTTP adapter interface

There are three methods declared in the MS-ASHTTP adapter interface IMS\_ASHTTPAdapter.

* These methods correspond to the two MS-ASHTTP commands: HTTP POST and HTTP OPTIONS. The other method ConfigureRequestPrefixFields is a customized method.

The methods are described in the following table.

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1 | HTTPPOST | This method is used to send HTTP POST request to the server and get the response. |
| 2 | HTTPOPTIONS | This method is used to send HTTP OPTIONS request to the server and get the response. |
| 3 | ConfigureRequestPrefixFields | This method is used to configure the fields in request line or request headers besides the command name and command parameters. |

MS-ASHTTP adapter interface methods

SUT control adapters

SUT control adapter interface

There is one method declared in the SUT control adapter interface IMS\_ASHTTPSUTControlAdapter.

The method is described in the following table.

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1 | ConfigureSSLSetting | This method is used to configure the SSL setting in SUT. |

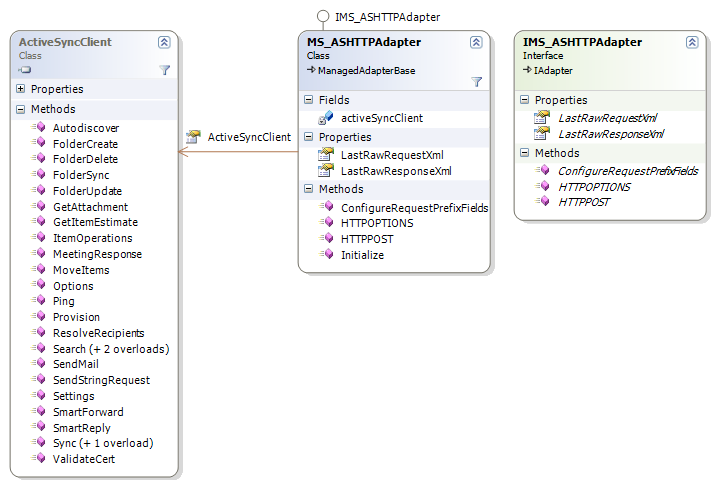
SUT control adapter interface method

### Adapter details

#### Protocol adapter

##### MS-ASHTTP protocol adapter

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



MS-ASHTTP adapter and ActiveSyncClient class diagram

The following outlines details of the class diagram:

Adapter interface

* IMS\_ASHTTPAdapter is the interface of the protocol adapter.
* IMS\_ASHTTPAdapter defines the methods invoked by test cases, including HTTPPOST, HTTPOPTIONS and ConfigureRequestPrefixFields methods.

Adapter implementation

* MS\_ASHTTP is the protocol adapter class of the test suite. It is used to implement IMS\_ASHTTPAdapter.
* The two properties LastRawRequestXml and LastRawResponseXml are used to get the raw request and raw response.
* The initialize method is used to initialize an instance of ActiveSyncClient.

Other class

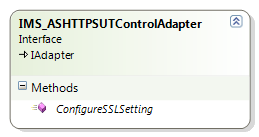
* ActiveSyncClient is used by MS\_ASHTTPAdapter to send (Autodiscover, FolderCreate, FolderDelete, FolderSync, FolderUpdate, GetAttachment, GetItemEstimate, ItemOperations, MeetingResponse, MoveItems, Ping, Provision, ResolveRecipients, Search, SendMail, Settings, SmartForward, SmartReply, Sync, ValidateCert, Options and SendStringRequest) command requests and retrieve command responses.

#### SUT control adapter

The following outlines details of class diagram:

##### SUT control adapter

The following figure shows the class diagram of the SUT control adapter.



SUT control adapter class diagram

The following outlines details of the class diagram:

* The IMS\_ASHTTPSUTControlAdapter is the interface of the SUT control adapter which is implemented by Microsoft PowerShell script. The implementation can be substituted by other implementation for the third party’s need.

## Test scenarios

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

|  |  |
| --- | --- |
| 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-ASHTTP scenarios

### S01\_HTTPPOSTPositive

Description

Test the positive behaviors issued by HTTP POST command.

Commands

* HTTP POST

Prerequisites

* Set the URI (Uniform Resource Identifier), HTTP version and query value type.

### S02\_HTTPPOSTNegative

Description

Test the negative behaviors issued by HTTP POST command.

Commands

* HTTP POST

Prerequisites

* Set the URI, HTTP version and query value type.

### S03\_HTTPPOSTOptionalHeader

Description

Test optional request header of HTTP POST command.

Commands

* HTTP POST

Prerequisites

* Set the URI, HTTP version and query value type.

### S04\_HTTPOPTIONSMessage

Description

Test HTTP OPTIONS command.

Commands

* HTTP OPTIONS

Prerequisites

* Set the URI, HTTP version and query value type.

## 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 23 traditional test cases are designed to cover the four scenarios mentioned in section [2.6 Test scenarios](#_Test_scenarios). Details of 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\_HTTPPOSTPositive | MSASHTTP\_S01\_TC01\_VerifyContentTypeResponseHeader |
| MSASHTTP\_S01\_TC02\_VerifyContentEncodingResponseHeader |
| MSASHTTP\_S01\_TC03\_CommandParameter\_AttachmentName\_Base64 |
| MSASHTTP\_S01\_TC04\_CommandParameter\_AttachmentName\_PlainText |
| MSASHTTP\_S01\_TC05\_CommandParameter\_SaveInSent\_Base64 |
| MSASHTTP\_S01\_TC06\_CommandParameter\_SaveInSent\_PlainText |
| MSASHTTP\_S01\_TC07\_CommandParameter\_LongId\_Base64 |
| MSASHTTP\_S01\_TC08\_CommandParameter\_LongId\_PlainText |
| MSASHTTP\_S01\_TC09\_CommandParameter\_Occurrence\_Base64 |
| MSASHTTP\_S01\_TC10\_CommandParameter\_Occurrence\_PlainText |
| MSASHTTP\_S01\_TC11\_CommandCode\_FolderRelatedCommands |
| MSASHTTP\_S01\_TC12\_CommandCode\_ItemRelatedCommands |
| MSASHTTP\_S01\_TC13\_CommandCode\_UserRelatedCommands |
| MSASHTTP\_S01\_TC14\_CommandCode\_ValidateCert |
| S02\_HTTPPOSTNegative | MSASHTTP\_S02\_TC01\_Verify400StatusCode |
| MSASHTTP\_S02\_TC02\_Verify401StatusCode |
| MSASHTTP\_S02\_TC03\_Verify403StatusCode |
| MSASHTTP\_S02\_TC04\_Verify404StatusCode |
| MSASHTTP\_S02\_TC05\_Verify501StatusCode |
| S03\_HTTPPOSTOptionalHeader | MSASHTTP\_S03\_TC01\_SetASAcceptMultiPartRequestHeader |
| MSASHTTP\_S03\_TC02\_SetUserAgentRequestHeader |
| MSASHTTP\_S03\_TC03\_SetPolicyKeyRequestHeader |
| S04\_HTTPOPTIONSMessage | MSASHTTP\_S04\_TC01\_HTTPOPTIONS |

Test case scenarios distribution

### Test case description

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

|  |  |
| --- | --- |
| Common cleanup | The client calls HTTPPOST with Sync command to delete the items created in test cases. |

Test case common prerequisites

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\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC01\_VerifyContentTypeResponseHeader |
| Description | This test case is intended to validate the Content-Type response header. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy by setting Content-Type Header to “application/vnd.ms-sync.wbxml”. |
| Cleanup | N/A |

MSASHTTP\_S01\_TC01\_VerifyContentTypeResponseHeader

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC02\_VerifyContentEncodingResponseHeader |
| Description | This test case is intended to validate the Content-Encoding response header. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy without setting AcceptEncoding header.  2. The client calls ConfigureRequestPrefixFields to set the AcceptEncoding header to "gzip".  3. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  4. The client calls ConfigureRequestPrefixFields to reset the AcceptEncoding header. |
| Cleanup | N/A |

MSASHTTP\_S01\_TC02\_VerifyContentEncodingResponseHeader

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC03\_CommandParameter\_AttachmentName\_Base64 |
| Description | This test case is intended to validate the AttachmentName command parameter with Base64 encoding query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls FolderSync command to synchronizes the collection hierarchy.  3. The client calls HTTPPOST with SendMail command to send an email with attachment to User2.  4. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  7. The client calls HTTPPOST with GetAttachment command to retrieve the email attachment.  8. The client calls ConfigureRequestPrefixFields to reset the query value type and credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC03\_CommandParameter\_AttachmentName\_Base64

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC04\_CommandParameter\_AttachmentName\_PlainText |
| Description | This test case is intended to validate the AttachmentName command parameter with Plain Text query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Plain text.  2. The client calls FolderSync command to synchronizes the collection hierarchy.  3. The client calls HTTPPOST with SendMail command to send an email with attachment to User2.  4. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  7. The client calls HTTPPOST with GetAttachment command to retrieve the email attachment.  8. The client calls ConfigureRequestPrefixFields to reset the query value type and credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC04\_CommandParameter\_AttachmentName\_PlainText

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC05\_CommandParameter\_SaveInSent\_Base64 |
| Description | This test case is intended to validate the SaveInSent, CollectionId and ItemId command parameters with Base64 encoding query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder.  4. The client calls HTTPPOST with SendMail command to send an email messages to User2 by setting command parameter SaveInSent to “1”.  5. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  6. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  7. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  8. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folders to get the received email.  9. The client calls HTTPPOST with SmartForward command to forward the email received in Inbox folder to User3 by setting command parameter SaveInSent to “1”.  10. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  11. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  12. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  13. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  14. The client calls HTTPPOST with SmartReply command to reply to the email received in Inbox folder to User2 by setting command parameter SaveInSent to “1”.  15. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  16. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  17. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  18. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folders to get the received email.  19. The client calls ConfigureRequestPrefixFields to reset the query value type and user credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC05\_CommandParameter\_SaveInSent\_Base64

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC06\_CommandParameter\_SaveInSent\_PlainText |
| Description | This test case is intended to validate the SaveInSent, CollectionId and ItemId command parameters with Plain Text query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Plain text.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder.  4. The client calls HTTPPOST with SendMail command to send an email messages to User2 by setting command parameter SaveInSent to “T”.  5. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  6. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  7. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  8. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folders to get the received email.  9. The client calls HTTPPOST with SmartForward command to forward the email received in Inbox folder to User3 by setting command parameter SaveInSent to “F”.  10. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  11. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  12. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  13. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  14. The client calls HTTPPOST with SmartReply command to reply to the email received in Inbox folder to User2 by setting command parameter SaveInSent to null.  15. The client calls HTTPPOST with Sync command to synchronize changes in Sent Items folder to get the saved email.  16. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  17. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  18. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folders to get the received email.  19. The client calls ConfigureRequestPrefixFields to reset the query value type and user credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC06\_CommandParameter\_SaveInSent\_PlainText

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC07\_CommandParameter\_LongId\_Base64 |
| Description | This test case is intended to validate the LongId command parameter with Base64 encoding query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls FolderSync command to synchronizes the collection hierarchy.  3. The client calls HTTPPOST with SendMail command to send an email messages to User2.  4. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls HTTPPOST with Search command to find the received email in Inbox folder.  7. The client calls HTTPPOST with SmartForward command to forward the received email to User3 by setting LongId command parameter returned by Search command.  8. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  9. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  10. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  11. The client calls ConfigureRequestPrefixFields to reset the query value type and user credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC07\_CommandParameter\_LongId\_Base64

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC08\_CommandParameter\_LongId\_PlainText |
| Description | This test case is intended to validate the LongId command parameter with Plain Text query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls FolderSync command to synchronizes the collection hierarchy.  3. The client calls HTTPPOST with SendMail command to send MIME-formatted email messages to User2.  4. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls HTTPPOST with Search command to find the received email in Inbox folder.  7. The client calls HTTPPOST with SmartForward command to forward the received email to User3 by setting LongId command parameter returned by Search command.  8. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  9. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  10. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  11. The client calls HTTPPOST with Search command to find the received email in Inbox folder.  12. The client calls HTTPPOST with SmartReply command to reply to the email received in Inbox folder to User2 by setting LongId command parameter returned by Search command.  13. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  14. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  15. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  16. The client calls ConfigureRequestPrefixFields to reset the query value type and user credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC08\_CommandParameter\_LongId\_PlainText

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC09\_CommandParameter\_Occurrence\_Base64 |
| Description | This test case is intended to validate the Occurrence command parameter with Base64 encoding query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  3. The client calls FolderSync command to synchronizes the collection hierarchy.  4. The client calls HTTPPOST with SendMail command to send a meeting request to User2.  5. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  6. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  7. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder and Calendar folder to get the received meeting request.  8. The client calls HTTPPOST with MeetingResponse command to accept the meeting request.  9. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder and Delete Items folder to get the accepted meeting request.  10. The client calls HTTPPOST with SmartForward command to forward the accepted meeting request to User1 by setting the Occrrence command parameter.  11. The client calls ConfigureRequestPrefixFields to switch the current user to User1.  12. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  13. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received meeting request.  14. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  15. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  16. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the meeting forward notification email.  17. The client calls ConfigureRequestPrefixFields to reset the query value type and credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC09\_CommandParameter\_Occurrence\_Base64

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC10\_CommandParameter\_Occurrence\_PlainText |
| Description | This test case is intended to validate the Occurrence command parameter with Plain Text query value type. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Plain text.  2. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  3. The client calls FolderSync command to synchronizes the collection hierarchy.  4. The client calls HTTPPOST with SendMail command to send a meeting request to User2.  5. The client calls ConfigureRequestPrefixFields to switch the current user to User2  6. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  7. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder and Calendar folder to get the received meeting request.  8. The client calls HTTPPOST with SmartReply command to reply to the meeting request to User3 with setting Occurrence command parameter.  9. The client calls ConfigureRequestPrefixFields to switch the current user to User3.  10. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  11. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received meeting request.  12. The client calls ConfigureRequestPrefixFields to reset the query value type and credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC10\_CommandParameter\_Occurrence\_PlainText

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC11\_CommandCode\_FolderRelatedCommands |
| Description | This test case is intended to validate the FolderSync, FolderCreate, FolderUpdate and FolderDelete command codes. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  2. The client calls HTTPPOST with FolderCreate command to create a new folder.  3. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  4. The client calls HTTPPOST with Sync command to synchronize changes in the new created folder.  5. The client calls HTTPPOST with FolderUpdate command to update the folder’s name and move the created folder to Sent Items folder.  6. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  7. The client calls HTTPPOST with FolderDelete command to delete the updated folder.  8. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC11\_CommandCode\_FolderRelatedCommands

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC12\_CommandCode\_ItemRelatedCommands |
| Description | This test case is intended to validate the Ping, MoveItems, GetItemEstimate and ItemOperations command codes. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Base64.  2. The client calls HTTPPOST with SendMail command to send MIME-formatted email messages to User2.  3. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  4. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  5. The client calls HTTPPOST with Ping command for changes of inbox folder.  6. The client calls HTTPPOST with Sync command to synchronize changes to get the received email.  7. The client calls HTTPPOST with FolderCreate command to create a new sub folder under Inbox folder.  8. The client calls HTTPPOST with MoveItems command to move the received email from Inbox folder to the new created folder.  9. The client calls HTTPPOST with Sync command to synchronize changes in the new created folder to get the moved email.  10. The client calls HTTPPOST with ItemOperations command to fetch the moved email with AcceptMultiPart command parameter.  11. The client calls HTTPPOST with FolderDelete command to delete the created folder.  12. The client calls ConfigureRequestPrefixFields to reset the query value type and user credential. |
| Cleanup | Common cleanup |

MSASHTTP\_S01\_TC12\_CommandCode\_ItemRelatedCommands

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC13\_CommandCode\_UserRelatedCommands |
| Description | This test case is intended to validate the ResolveRecipients and Settings command codes. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTPPOST with ResolveRecipients command to resolve a supplied recipient.  2. The client calls HTTPPOST with Settings command. |
| Cleanup | N/A |

MSASHTTP\_S01\_TC13\_CommandCode\_ UserRelatedCommands

|  |  |
| --- | --- |
| S01\_HTTPPOSTPositive | |
| Test case ID | MSASHTTP\_S01\_TC14\_CommandCode\_ValidateCert |
| Description | This test case is intended to validate the ValidateCert command code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTPPOST with ValidateCert command. |
| Cleanup | N/A |

MSASHTTP\_S01\_TC14\_CommandCode\_ValidateCert

|  |  |
| --- | --- |
| S02\_HTTPPOSTNegative | |
| Test case ID | MSASHTTP\_S02\_TC01\_Verify400StatusCode |
| Description | This test case is intended to validate the 400 Bad Request HTTP POST status code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the ActiveSync protocol version to “191”.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls ConfigureRequestPrefixFields to reset the ActiveSync protocol version.  4. The client calls ConfigureRequestPrefixFields to change the ActiveSync protocol version to “191”.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls ConfigureRequestPrefixFields to reset the ActiveSync protocol version. |
| Cleanup | N/A |

MSASHTTP\_S02\_TC01\_Verify400StatusCode

|  |  |
| --- | --- |
| S02\_HTTPPOSTNegative | |
| Test case ID | MSASHTTP\_S02\_TC02\_Verify401StatusCode |
| Description | This test case is intended to validate the 401 Unauthorized HTTP POST status code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to remove the user credential.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls ConfigureRequestPrefixFields to reset the credential.  4. The client calls ConfigureRequestPrefixFields to change the authentication to User1 with an invalid password.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls ConfigureRequestPrefixFields to reset the credential. |
| Cleanup | N/A |

MSASHTTP\_S02\_TC02\_Verify401StatusCode

|  |  |
| --- | --- |
| S02\_HTTPPOSTNegative | |
| Test case ID | MSASHTTP\_S02\_TC03\_Verify403StatusCode |
| Description | This test case is intended to validate the 403 Forbidden HTTP POST status code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls SUT control adapter to enable the SSL setting of SUT.  2. The client calls ConfigureRequestPrefixFields to change the prefix of URI to “HTTP”.  3. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  4. The client calls ConfigureRequestPrefixFields to reset the prefix of URI.  5. The client calls SUT control adapter to disable the SSL setting of SUT. |
| Cleanup | N/A |

MSASHTTP\_S02\_TC03\_Verify403StatusCode

|  |  |
| --- | --- |
| S02\_HTTPPOSTNegative | |
| Test case ID | MSASHTTP\_S02\_TC04\_Verify404StatusCode |
| Description | This test case is intended to validate the 404 Not Found HTTP POST status code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to update the URI to an invalid value.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls ConfigureRequestPrefixFields to reset the URI. |
| Cleanup | N/A |

MSASHTTP\_S02\_TC04\_Verify404StatusCode

|  |  |
| --- | --- |
| S02\_HTTPPOSTNegative | |
| Test case ID | MSASHTTP\_S02\_TC05\_Verify501StatusCode |
| Description | This test case is intended to validate the 501 Not Implemented HTTP POST status code. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Plain text.  2. The client calls HTTPPOST with NotExist command.  3. The client calls ConfigureRequestPrefixFields to reset the query value type. |
| Cleanup | N/A |

MSASHTTP\_S02\_TC05\_Verify501StatusCode

|  |  |
| --- | --- |
| S03\_HTTPPOSTOptionalHeader | |
| Test case ID | MSASHTTP\_S03\_TC01\_SetASAcceptMultiPartRequestHeader |
| Description | This test case is intended to validate the MS-ASAcceptMultiPart optional header in HTTP POST request. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to change the query value type to Plain text.  2. The client calls FolderSync command to synchronizes the collection hierarchy.  3. The client calls HTTPPOST with SendMail command to send email to User2.  4. The client calls ConfigureRequestPrefixFields to switch the current user to User2.  5. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  6. The client calls HTTPPOST with Sync command to synchronize changes in Inbox folder to get the received email.  7. The client calls HTTPPOST with ItemOperations command to fetch the received email with MS-ASAcceptMultiPart request header is set to “T”.  8. The client calls HTTPPOST with ItemOperations command to fetch the received email with MS-ASAcceptMultiPart request header is set to “F”.  9. The client calls HTTPPOST with ItemOperations command to fetch the received email without setting MS-ASAcceptMultiPart request header.  10. The client calls ConfigureRequestPrefixFields to reset the query value type. |
| Cleanup | Common cleanup |

MSASHTTP\_S03\_TC01\_SetASAcceptMultiPartRequestHeader

|  |  |
| --- | --- |
| S03\_HTTPPOSTOptionalHeader | |
| Test case ID | MSASHTTP\_S03\_TC02\_SetUserAgentRequestHeader |
| Description | This test case is intended to validate the User-Agent optional header in HTTP POST request. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to set the User-Agent request header to “ASOM”.  2. The client calls HTTPPOST with FolderSync command to synchronize the collection hierarchy.  3. The client calls ConfigureRequestPrefixFields to reset the User-Agent request header. |
| Cleanup | N/A |

MSASHTTP\_S03\_TC02\_SetUserAgentRequestHeader

|  |  |
| --- | --- |
| S03\_HTTPPOSTOptionalHeader | |
| Test case ID | MSASHTTP\_S03\_TC03\_SetPolicyKeyRequestHeader |
| Description | This test case is intended to validate the X-MS-PolicyKey optional header and Policy key optional field in HTTP POST request. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls ConfigureRequestPrefixFields to set the query value type to PlainText.  2. The client calls HTTPPOST with Provision command to download the policy settings.  3. The client calls HTTPPOST with Provision command to acknowledge the policy settings with setting X-MS-PolicyKey request header of the PlainText encoded query value to the policy key value returned from step 1.  4. The client calls ConfigureRequestPrefixFields to set the query value type to Base64.  5. The client calls HTTPPOST with Provision command to download the policy settings.  6. The client calls HTTPPOST with Provision command to acknowledge the policy settings by setting Policy key field of the base64 encoded query value to policy key value returned from step 5.  7. The client calls ConfigureRequestPrefixFields to reset the query value type. |
| Cleanup | N/A |

MSASHTTP\_S03\_TC03\_SetPolicyKeyRequestHeader

|  |  |
| --- | --- |
| S04\_HTTPOPTIONSMessage | |
| Test case ID | MSASHTTP\_S04\_TC01\_HTTPOPTIONS |
| Description | This test case is intended to validate the HTTP OPTIONS command. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The client calls HTTP OPTIONS command to send HTTP OPTIONS request to the server and get the response. |
| Cleanup | N/A |

MSASHTTP\_S04\_TC01\_HTTPOPTIONS