

MS-ASTASK Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how the MS-ASTASK test suite is designed to test the MS-ASTASK 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](#_Toc387412244)

[1.1 Configuring the test suite client 3](#_Toc387412245)

[1.1.1 Configuring the test suite client manually 3](#_Toc387412246)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc387412247)

[1.2 Configuring the system under test (SUT) 3](#_Toc387412248)

[1.2.1 Configuring the SUT manually 3](#_Toc387412249)

[1.2.2 Configuring the SUT by scripts 3](#_Toc387412250)

[1.3 Configuring the SHOULD/MAY requirements 3](#_Toc387412251)

[2 Test suite design 5](#_Toc387412252)

[2.1 Assumptions, scope and constraints 5](#_Toc387412253)

[2.2 Test suite architecture 5](#_Toc387412254)

[2.3 Technical dependencies and considerations 7](#_Toc387412255)

[2.4 Adapter design 7](#_Toc387412256)

[2.4.1 Adapter overview 7](#_Toc387412257)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387412258)

[2.4.3 Adapter abstract layer 8](#_Toc387412259)

[2.4.4 Adapter details 8](#_Toc387412260)

[2.5 Test scenarios 10](#_Toc387412261)

[2.5.1 S01\_SyncCommand 10](#_Toc387412262)

[2.5.2 S02\_ItemOperationsCommand 10](#_Toc387412263)

[2.5.3 S03\_SearchCommand 11](#_Toc387412264)

[2.6 Test case design 12](#_Toc387412265)

[2.6.1 Traditional test case design 12](#_Toc387412266)

[2.6.2 Test case description 12](#_Toc387412267)

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

1. Open MS-ASTASK\TestSuite\MS-ASTASK\_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="UserName" value="MSASTASK\_User01"
* Property name="Password" 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-ASTASK\_ExchangeServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-ASTASK\_ExchangeServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-ASTASK\_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-ASTASK\_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 in ActiveSync command response.
* 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: 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-ASTASK test suite architecture

* SUT hosts the task 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-ASTASK 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-ASTASK Open Specification.
* Test cases use the MS-ASTASK protocol adapter to call and get the results of the MS-ASTASK methods.
* MS-ASTASK protocol adapter is used in the test cases. The test cases call the methods in the interface to invoke the MS-ASTASK protocol adapter’s methods.
* MS-ASTASK protocol adapter uses ActiveSyncClient to send command request and retrieve command response.
* ActiveSyncClient encodes and decodes Sync command, Search command, ItemOperations command and FolderSync 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 Protocol Test Framework (PTF) to derive managed adapter.
* This test suite depends on MS-ASWBXML to encode XMLrequests bodies into WBXML for transmission to an ActiveSync server.
* This test suite depends on MS-ASHTTP to synchronize the data which is stored on the server.
* This test suite depends on the xsd.exe tool in the .NET Framework SDK to generate structures used in the MS-ASTASK request and response.

Encryption consideration

* Transportation of MS-ASTASK 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-ASTASK protocol adapter
* The MS-ASTASK adapter is a managed adapter, which is derived from the ManagedAdapterBase class in PTF.
* The MS-ASTASK adapter has the following functionalities
* Choose HTTP or HTTPS for transport;
* Construct requests of 3 MS-ASTASK commands: ItemOperations, Search and Sync;
* 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-ASTASK adapter gets the parameter values and calls the corresponding commands in ActiveSyncClient. The ActiveSyncClient serializes the parameter values to XML elements, and then encodes requests bodies into WBXML for transmission to an ActiveSync server.

Message consumption

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

### Adapter abstract layer

Protocol adapter

MS-ASTASK adapter interface

There are five methods declared in the MS-ASTASK adapter interface IMS\_ASTASKAdapter.

The methods ItemOperations, Search and Sync correspond to the three commands used by MS-ASTASK. The method FolderSync is used to synchronize the collection hierarchy, and the method SendStringRequest is used to send a string request.

The methods are described in the following table.

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1 | FolderSync | Synchronize the collection hierarchy. |
| 2 | ItemOperations | Fetch all information about Exchange objects. |
| 3 | Search | Retrieve Task class items from the server that match the criteria specified by the client. |
| 4 | Sync | Synchronize Task items for a specified user with the existing e-mail items stored on the server. |
| 5 | SendStringRequest | Send a string request and get a response from server. |

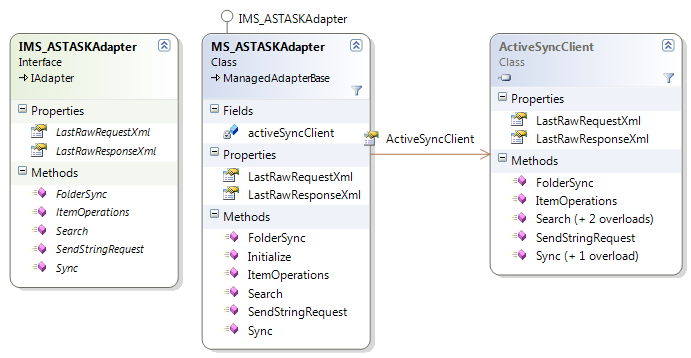
MS-ASTASK adapter interface methods

### Adapter details

#### Protocol adapter

##### MS-ASTASK protocol adapter

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



MS-ASTASK adapter and ActiveSyncClient class diagrams

The following outlines details of the class diagrams:

Adapter interface

* IMS\_ASTASKAdapter is the interface of the protocol adapter.
* IMS\_ASTASKAdapter defines the methods invoked by the test cases, including the following five methods: FolderSync, SendStringRequest, ItemOperations, Search and Sync.

Adapter implementation

* MS\_ASTASKAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_ASTASKAdapter.
* MS\_ASTASKAdapter invokes the methods defined in ActiveSyncClient to operate with Task class data between client and server.
* The two properties LastRawRequestXml and LastRawResponseXml are used to get the raw request and raw response xml data.

## Test scenarios

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

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

MS-ASTASK scenarios

### S01\_SyncCommand

Description

This scenario is to test Task class element on the server by using Sync command.

Commands

* Sync

Prerequisites

Set HTTP or HTTPS as the transport.

Cleanup

Call Sync command with SyncCollectionDelete element to delete the items created on the server.

### S02\_ItemOperationsCommand

Description

This scenario is to test Task class element on the server by using ItemOperations command.

Commands

* ItemOperations

Prerequisites

Set HTTP or HTTPS as the transport.

Cleanup

Call Sync command with SyncCollectionDelete element to delete the items created on the server.

### S03\_SearchCommand

Description

This scenario is to test Task class element on the server by using Search command.

Commands

* Search

Prerequisites

Set HTTP or HTTPS as the transport.

Cleanup

Call Sync command with SyncCollectionDelete element to delete the items created on the server.

## 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 27 traditional test cases designed to cover the scenario 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\_SyncCommand | MSASTASK\_S01\_TC01\_CreateTaskItemRecursDaily |
| MSASTASK\_S01\_TC02\_CreateTaskItemRecursDailyWithoutStart |
| MSASTASK\_S01\_TC03\_CreateTaskItemFailWithDayOfMonth |
| MSASTASK\_S01\_TC04\_CreateTaskItemRecursMonthly |
| MSASTASK\_S01\_TC05\_CreateTaskItemRecursYearly |
| MSASTASK\_S01\_TC06\_CreateTaskItemRecursWeekly |
| MSASTASK\_S01\_TC07\_CreateTaskItemFailWithDayOfWeek |
| MSASTASK\_S01\_TC08\_CreateTaskItemRecursWeeklyWithInvalidFirstDayOfWeek |
| MSASTASK\_S01\_TC09\_CreateTaskItemWithInvalidImportanceElement |
| MSASTASK\_S01\_TC10\_CreateTaskItemRecursYearlyOnTheNthDay |
| MSASTASK\_S01\_TC11\_CreateTaskItemWithInvalidReminderSetElement |
| MSASTASK\_S01\_TC12\_CreateTaskItemRecursMonthlyOnTheNthDay |
| MSASTASK\_S01\_TC13\_CreateTaskItemFailWithWeekOfMonth |
| MSASTASK\_S01\_TC14\_CreateTaskItemFailWithCalendarType |
| MSASTASK\_S01\_TC15\_CreateTaskItemOccurrencesAndUntilBothSet |
| MSASTASK\_S01\_TC16\_CreateTaskItemRecurrenceWithoutType |
| MSASTASK\_S01\_TC17\_CreateTaskItemWithDateCompletedElement |
| MSASTASK\_S01\_TC18\_CreateAndChangeSubject |
| MSASTASK\_S01\_TC19\_CreateTaskItemCategoriesWithoutChildElements |
| MSASTASK\_S01\_TC20\_CreateTaskItemRecursMonthlyWithCalendarTypeReturned |
| MSASTASK\_S01\_TC21\_CreateTaskItemRecursMonthlyOnTheNthDayWithCalendarTypeReturned |
| MSASTASK\_S01\_TC22\_CreateTaskItemRecursYearlyWithCalendarTypeReturned |
| MSASTASK\_S01\_TC23\_CreateTaskItemRecursYearlyOnTheNthDayWithCalendarTypeReturned |
| MSASTASK\_S01\_TC24\_CreateTaskItemRecursWeeklyWithFirstDayOfWeekReturned |
| S02\_ItemOperationsCommand | MSASTASK\_S02\_TC01\_RetrieveTaskItemWithItemOperations |
| S03\_SearchCommand | MSASTASK\_S03\_TC01\_RetrieveTaskItemWithSearch |
| MSASTASK\_S03\_TC02\_RetrieveInvalidTaskItemWithSearch |

Test case scenario distribution

### Test case description

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

|  |  |
| --- | --- |
| Common prerequisites | 1. Uses HTTP or HTTPS as the transport. 2. Call FolderSync to get ServerId of Task Folder. |
| Common cleanup | Call Sync command with SyncCollectionDelete element to delete the items created on the server. |

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\_SyncCommand | |
| Test case ID | MSASTASK\_S01\_TC01\_CreateTaskItemRecursDaily |
| Description | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is 'recurs daily'. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call Sync command with with airsync:Add element in the request message to add a task which contains a Recurrence whose Type element is “recurs daily” into the Task Folder. 2. Call Sync command to get the Task item. |
| Cleanup | Common cleanup |

MSASTASK\_S01\_TC01\_CreateTaskItemRecursDaily

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC02\_CreateTaskItemRecursDailyWithoutStart |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is 'recurs daily', but without Start element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs dayly” and with “DayOfMonth” element, but without “Start” element into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC02\_CreateTaskItemRecursDailyWithoutStart

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC03\_CreateTaskItemFailWithDayOfMonth |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is not 2 or 5 and contains a 'DayOfMonth' element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs daily” and with “DayOfMonth” element into the Task Folder. 2. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs weekly” and with “DayOfMonth” element into the Task Folder. 3. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs monthly on the nth day” and with “DayOfMonth” element into the Task Folder. 4. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs yearly on the nth day” and with “DayOfMonth” element into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC03\_CreateTaskItemFailWithDayOfMonth

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC04\_CreateTaskItemRecursMonthly |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence of which the Type element is 'recurs monthly'. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence of which the Type element is “recurs monthly” into the Task Folder. 2. Call Sync command to get the Task item. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC04\_CreateTaskItemRecursMonthly

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC05\_CreateTaskItemRecursYearly |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence of which the Type element is 'recurs yearly'. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request to add a task with Recurrence whose Type element is “”recurs yearly” into the Task Folder. 2. Call Sync command to get the Task item. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC05\_CreateTaskItemRecursYearly

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC06\_CreateTaskItemRecursWeekly |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is "recurs weekly". |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task which contains a Recurrence whose Type element is “recurs weekly” into the Task Folder. 2. Call Sync command to get the Task item. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC06\_CreateTaskItemRecursWeekly

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC07\_CreateTaskItemFailWithDayOfWeek |
| **Description** | This test case is designed to verify the requirements about processing tasks with a Recurrence whose Type element is not 0, 1, 3 and 6 and contains a 'DayOfWeek' element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with a Recurrence whose Type element is “recurs monthly” and contains a “DayOfWeek” element into the Task Folder. 2. Call Sync command with airsync:Add element in the request message to add a task with a Recurrence whose Type element is “recurs yearly” and contains a “DayOfWeek” element into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC07\_CreateTaskItemFailWithDayOfWeek

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC08\_CreateTaskItemRecursWeeklyWithInvalidFirstDayOfWeek |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is 'recurs weekly' and contains an invalid 'FirstDayOfWeek' value. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs weekly” and contains an invalid “FirstDayOfWeek” value into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC08\_CreateTaskItemRecursWeeklyWithInvalidFirstDayOfWeek

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC09\_CreateTaskItemWithInvalidImportanceElement |
| **Description** | This test case is designed to verify the requirements about processing tasks with an invalid 'Importance' value. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with an invalid “Importance” value into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC09\_CreateTaskItemWithInvalidImportanceElement

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC10\_CreateTaskItemRecursYearlyOnTheNthDay |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type element is 'recurs yearly on the Nth day'. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs yearly on the Nth day” into the Task Folder. 2. Call Sync command to get the Task item. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC10\_CreateTaskItemRecursYearlyOnTheNthDay

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC11\_CreateTaskItemWithInvalidReminderSetElement |
| **Description** | This test case is designed to verify the requirements about processing tasks with an invalid 'ReminderSet' value. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with an invalid “ReminderSet” value into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC11\_CreateTaskItemWithInvalidReminderSetElement

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC12\_CreateTaskItemRecursMonthlyOnTheNthDay |
| **Description** | This test case is designed to verify the requirements about processing task with Recurrence whose Type element is 'recurs monthly on the Nth day'. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose Type element is “recurs monthly on the Nth day” into the Task Folder. 2. Call Sync command to get the Task item. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC12\_CreateTaskItemRecursMonthlyOnTheNthDay

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC13\_CreateTaskItemFailWithWeekOfMonth |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type value is not 3 or 6 and contains a 'WeekOfMonth' element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs daily” and contains a “WeekOfMonth” element into the Task Folder. 2. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs weekly” and contains a “WeekOfMonth” element into the Task Folder. 3. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs monthly” and contains a “WeekOfMonth” element into the Task Folder. 4. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs yearly” and contains a “WeekOfMonth” element into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC13\_CreateTaskItemFailWithWeekOfMonth

|  |  |
| --- | --- |
| S01\_SyncCommand | |
| Test case ID | MSASTASK\_S01\_TC14\_CreateTaskItemFailWithCalendarType |
| Description | This test case is designed to verify the requirements about processing tasks with Recurrence whose type value is not 2, 3, 5 or 6 and contains a 'CalendarType' element. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs daily” and contains a “CalendarType” element into the Task Folder. 2. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs weekly” and contains a “CalendarType” element into the Task Folder. |
| Cleanup | Common cleanup |

MSASTASK\_S01\_TC14\_CreateTaskItemFailWithCalendarType

|  |  |
| --- | --- |
| S01\_SyncCommand | |
| Test case ID | MSASTASK\_S01\_TC15\_CreateTaskItemOccurrencesAndUntilBothSet |
| Description | This test case is designed to verify the Recurrence element when Occurrence and Until elements are both set. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call Sync method with an Add element to add a task, with Occurrences and Until elements both set, to the server. 2. Call Sync method to get the task added in step 1 from server. |
| Cleanup | Common cleanup |

MSASTASK\_S01\_TC15\_CreateTaskItemOccurrencesAndUntilBothSet

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC16\_CreateTaskItemRecurrenceWithoutType |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence without Type element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whout Type element into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC16\_CreateTaskItemRecurrenceWithoutType

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC17\_CreateTaskItemWithDateCompletedElement |
| **Description** | This test case is designed to verify the requirements about processing tasks with a 'DateCompleted' element. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with a “Completed” element set as “1” into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC17\_CreateTaskItemWithDateCompletedElement

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC18\_CreateAndChangeSubject |
| **Description** | This test case is designed to verify the requirements about not including ReminderSet and importance element into airsync:Change. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. 3. Call Sync command with aiysync:Change element in the request message to update the subject of the task added in step 1. 4. Call Sync method to get the task updated in step 3 from server. 5. Call Sync command with airsync:Add element in the request message to add a task into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC18\_CreateAndChangeSubject

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC19\_CreateTaskItemCategoriesWithoutChildElements |
| **Description** | This test case is designed to verify the requirements about processing tasks Categories element that doesn't contain child elements. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command request with airsync:Add element in the request message to add a task with “Categories” element but without child elements into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC19\_CreateTaskItemCategoriesWithoutChildElements

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC20\_CreateTaskItemRecursMonthlyWithCalendarTypeReturned |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type value is 2 and the CalendarType is returned in Sync response. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs monthly” and contains a “CalendarType” element into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC20\_CreateTaskItemRecursMonthlyWithCalendarTypeReturned

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC21\_CreateTaskItemRecursMonthlyOnTheNthDayWithCalendarTypeReturned |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type value is 3 and the CalendarType is returned in Sync response. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs monthly on the nth day” and contains a “CalendarType” element into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC21\_CreateTaskItemRecursMonthlyOnTheNthDayWithCalendarTypeReturned

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC22\_CreateTaskItemRecursYearlyWithCalendarTypeReturned |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type value is 5 and the CalendarType is returned in Sync response. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs yearly” and contains a “CalendarType” element into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC22\_CreateTaskItemRecursYearlyWithCalendarTypeReturned

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC23\_CreateTaskItemRecursYearlyOnTheNthDayWithCalendarTypeReturned |
| **Description** | This test case is designed to verify the requirements about processing tasks with Recurrence whose Type value is 6 and the CalendarType is returned in Sync response. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs yearly on the nth day” and contains a “CalendarType” element into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC23\_CreateTaskItemRecursYearlyOnTheNthDayWithCalendarTypeReturned

|  |  |
| --- | --- |
| [S01\_SyncCommand](#S01) | |
| **Test case ID** | MSASTASK\_S01\_TC24\_CreateTaskItemRecursWeeklyWithFirstDayOfWeekReturned |
| **Description** | This test case is designed to verify the requirements about processing tasks with FirstDayOfWeek returned in response. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command with airsync:Add element in the request message to add a task with Recurrence whose type element is “recurs weekly” into the Task Folder. 2. Call Sync method to get the task added in step 1 from server. |
| **Cleanup** | Common cleanup |

MSASTASK\_S01\_TC24\_CreateTaskItemRecursWeeklyWithFirstDayOfWeekReturned

|  |  |
| --- | --- |
| [S02\_ItemOperationsCommand](#S01) | |
| **Test case ID** | MSASTASK\_S02\_TC01\_RetrieveTaskItemWithItemOperations |
| **Description** | This test case is designed to verify the requirements about processing the ItemOperations command. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command request with airsync:Add element in the request message to add a task into the Task Folder. 2. Call Itemoperations command to fetch the task added in step 1. |
| **Cleanup** | Common cleanup |

MSASTASK\_S02\_TC01\_RetrieveTaskItemWithItemOperations

|  |  |
| --- | --- |
| [S03\_SearchCommand](#S01) | |
| **Test case ID** | MSASTASK\_S03\_TC01\_RetrieveTaskItemWithSearch |
| **Description** | This test case is designed to verify the requirements about processing the Search command. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command request with airsync:Add element in the request message to add a task into the Task Folder. 2. Call Search command to search the task into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S03\_TC01\_RetrieveTaskItemWithSearch

|  |  |
| --- | --- |
| [S03\_SearchCommand](#S01) | |
| **Test case ID** | MSASTASK\_S03\_TC02\_RetrieveInvalidTaskItemWithSearch |
| **Description** | This test case is designed to verify the requirements about processing the Search command with invalid Task item. |
| **Prerequisites** | Common prerequisites |
| **Test execution steps** | 1. Call Sync command request with airsync:Add element in the request message to add a task into the Task Folder. 2. Call Search command to search the task into the Task Folder. |
| **Cleanup** | Common cleanup |

MSASTASK\_S03\_TC02\_RetrieveInvalidTaskItemWithSearch