

MS-MEETS Test Suite Specification

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

Contents

[1 Configuring the test suite 4](#_Toc356306213)

[1.1 Configuring the test suite client 4](#_Toc356306214)

[1.1.1 Configuring the test suite client manually 4](#_Toc356306215)

[1.1.2 Configuring the test suite client by scripts 4](#_Toc356306216)

[1.2 Configuring the system under test (SUT) 4](#_Toc356306217)

[1.2.1 Configuring the SUT manually 4](#_Toc356306218)

[1.2.2 Configuring the SUT by scripts 5](#_Toc356306219)

[1.3 Configuring the SHOULD/MAY requirements 5](#_Toc356306220)

[2 Test suite design 5](#_Toc356306221)

[2.1 Assumptions, scope and constraints 5](#_Toc356306222)

[Assumptions 5](#_Toc356306223)

[Scope 5](#_Toc356306224)

[In scope 5](#_Toc356306225)

[Out of scope 5](#_Toc356306226)

[Constraints 6](#_Toc356306227)

[2.2 Test suite architecture 6](#_Toc356306228)

[2.3 Technical dependencies and encryption considerations 7](#_Toc356306229)

[Dependencies 7](#_Toc356306230)

[Encryption consideration 7](#_Toc356306231)

[2.4 Adapter design 7](#_Toc356306232)

[2.4.1 Adapter overview 7](#_Toc356306233)

[Protocol adapter 7](#_Toc356306234)

[SUT control adapter 7](#_Toc356306235)

[2.4.2 Technical feasibility of adapter approach 8](#_Toc356306236)

[Message generation 8](#_Toc356306237)

[Message consumption 8](#_Toc356306238)

[SUT control adapter 8](#_Toc356306239)

[2.4.3 Adapter abstract layer 8](#_Toc356306240)

[Protocol adapter 8](#_Toc356306241)

[MS-MEETS adapter interface 8](#_Toc356306242)

[SUT control adapter 8](#_Toc356306243)

[SUT control adapter interface 8](#_Toc356306244)

[2.4.4 Adapter details 8](#_Toc356306245)

[*2.4.4.1* Protocol adapter 8](#_Toc356306246)

[2.4.4.1.1 MS-MEETS protocol adapter 8](#_Toc356306247)

[Adapter interface 10](#_Toc356306248)

[Adapter implementation 10](#_Toc356306249)

[Other classes 10](#_Toc356306250)

[Enumeration 10](#_Toc356306251)

[*2.4.4.2* SUT control adapter 10](#_Toc356306252)

[2.4.4.2.1 MS-MEETS SUT control adapter 10](#_Toc356306253)

[2.5 Test scenarios 11](#_Toc356306254)

[2.5.1 S01\_MeetingWorkspace 11](#_Toc356306255)

[Description 11](#_Toc356306256)

[Operations 11](#_Toc356306257)

[Prerequisite 11](#_Toc356306258)

[Cleanup 11](#_Toc356306259)

[2.5.2 S02\_Meeting 11](#_Toc356306260)

[Description 11](#_Toc356306261)

[Operations 11](#_Toc356306262)

[Prerequisite 12](#_Toc356306263)

[Cleanup 12](#_Toc356306264)

[2.5.3 S03\_MeetingFromICal 12](#_Toc356306265)

[Description 12](#_Toc356306266)

[Operations 12](#_Toc356306267)

[Prerequisite 12](#_Toc356306268)

[Cleanup 12](#_Toc356306269)

[2.5.4 S04\_RecurringMeeting 12](#_Toc356306270)

[Description 12](#_Toc356306271)

[Operations 12](#_Toc356306272)

[Prerequisite 13](#_Toc356306273)

[Cleanup 13](#_Toc356306274)

[2.6 Test case design 14](#_Toc356306275)

[2.6.1 Traditional test case design 14](#_Toc356306276)

[2.6.2 Test case description 15](#_Toc356306277)

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

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

Property name=

"CommonConfigurationFileName" value="SharePointCommonConfiguration.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 first search through its specific configuration file and use the properties from there if they are defined, before looking for them from the common configuration file (if specified).

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

* Property name="TargetServiceUrl" value="[TransportType]://[SUTComputerName]/sites/[SiteCollectionName][EntryUrl]"
* Property name="SiteCollectionName" value="MSMEETS\_SiteCollection"
* Property name="EntryUrl" value="/\_vti\_bin/meetings.asmx"
* Property name="ServiceTimeOut" value="10"
* Property name="OrganizerEmail" value="administrator@contoso.com"
* Property name="AttendeeEmail" value="[attendee@contoso.com](mailto:attendee@contoso.com)"

### Configuring the test suite client by scripts

To configure the test suite client using scripts, see section 5.2.4 of the [SharePointTestSuiteDeploymentGuide.docx](../SharePointTestSuiteDeploymentGuide.docx).

## Configuring the system under test (SUT)

### Configuring the SUT manually

To manually configure the SUT, see section 5.1.3 of the [SharePointTestSuiteDeploymentGuide.docx](../SharePointTestSuiteDeploymentGuide.docx).

### Configuring the SUT by scripts

To configure the SUT using scripts, see section 5.1.2 of the [SharePointTestSuiteDeploymentGuide.docx.](../SharePointTestSuiteDeploymentGuide.docx)

## Configuring the SHOULD/MAY requirements

All the implementation of the SHOULD/MAY and endnote related requirements are pre-configured in the format "<Property name="RXXXEnabled" value="XXXX"/>" for six Microsoft product versions in six SHOULD/MAY PTFConfig files:

* MS-MEETS\_WindowsSharePointServices3\_SHOULDMAY.deployment.ptfconfig
* MS-MEETS\_SharePointServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-MEETS\_SharePointFoundation2010\_SHOULDMAY.deployment.ptfconfig
* MS-MEETS\_SharePointServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-MEETS\_SharePointFoundation2013\_SHOULDMAY.deployment.ptfconfig
* MS-MEETS\_SharePointServer2013\_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 SharePoint Foundation 2010 is chosen,user can open **MS-MEETS\_SharePointFoundation2010\_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 12 operations: AddMeeting, AddMeetingFromICal, CreateWorkspace, DeleteWorkspace, GetMeetingsInformation, GetMeetingWorkspaces, RemoveMeeting, RestoreMeeting, SetAttendeeResponse, SetWorkspaceTitle, UpdateMeeting and UpdateMeetingFromICal.
* This test suite will verify the Full WSDL which is provided in the Open Specification.
* This test suite will verify the server-side and testable requirements by running all the test cases on both HTTP and HTTPS.
* This test suite will verify operations over SOAP 1.1 and SOAP 1.2.

Out of scope

* This test suite will not verify the requirements related to the client behaviors.
* This test suite will not verify the requirements related to the 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-MEETS test suite architecture.

* SUT hosts the Meetings Web Service which this test suite runs against.
* SUT is the protocol server implementation from third-party user’s point of view.
* This test suite was used to test the MS-MEETS protocol implementation against the following SharePoint versions.
* Windows SharePoint Services 3.0 Service Pack 3 (SP3)
* Microsoft SharePoint Foundation 2010 Service Pack 2 (SP2)
* Microsoft SharePoint Foundation 2013 SP1
* Microsoft Office SharePoint Server 2007 Service Pack 3 (SP3)
* Microsoft SharePoint Server 2010 Service Pack 2 (SP2)
* Microsoft SharePoint Server 2013 SP1
* Test Suite acts as the client to communicate with the SUT and validates the requirements gathered from MS-MEETS Open Specification.
* The test cases invoke the MS-MEETS adapter to call the MS-MEETS operations and validate the response from SUT. Test cases also use the SUT control adapter to set the SUT to the test case specific situation.
* MS-MEETS adapter is used in the test cases. The test cases call the methods in the adapter interfaces to invoke the MS-MEETS operation.
* The test cases also use the SUT control adapter to set/modify the SUT environment by calling the methods in the SUT control adapter interface to configure the SUT.

## Technical dependencies and encryption considerations

Dependencies

* This test suite depends on the SOAP messaging protocol for exchanging structured data and type information.
* This test suite depends on HTTP protocol or HTTPS protocol to transmit the messages.
* This test suite depends on the wsdl.exe tool in the .NET Framework SDK to generate the MS-MEETS proxy class.
* This test suite depends on the Protocol Test Framework (PTF) to derive managed adapters and script adapters.

Encryption consideration

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

## Adapter design

### Adapter overview

This protocol consists of one protocol adapter and one SUT control adapter.

Protocol adapter

* MS-MEETS Adapter
* The MS-MEETS adapter is a managed adapter, which is derived from the ManagedAdapterBase class in the Protocol Test Framework (PTF).
* The MS-MEETS Adapter has the following functionalities
* Choose HTTP or HTTPS and SOAP 1.1 or 1.2 for transport.
* Construct requests of 12 MS-MEETS operations.
* Communicate with the SUT by sending requests to the SUT and receive the corresponding responses from the SUT.
* Parse the response messages and validate the messages according to the WSDL schema.
* Generate the result log.
* The MS-MEETS Adapter uses the C# proxy class, which is generated by running the wsdl.exe tool against the full WSDL of this protocol to send SOAP request messages and receive SOAP response messages. The wsdl.exe can be found in Microsoft .NET Framework SDK tools.

SUT control adapter

* The SUT control adapter will be a PowerShell script adapter; it is invoked by the test cases.
* The SUT control adapter has the following functionalities
* Remove any existing workspace on the specified site of the SUT before running any test case.
* Make sure interactive mode can be used.

### Technical feasibility of adapter approach

Message generation

The MS-MEETS adapter gets the parameter values of the WSDL operations and calls the corresponding operations in MS-MEETS proxy class, the MS-MEETS proxy class serializes the parameter values to XML elements to format the SOAP request messages, then the SOAP request messages are sent out by the MS-MEETS proxy class.

Message consumption

The messages received from the SUT will be parsed in the MS-MEETS proxy class and be passed upon to the MS-MEETS adapter. Then these messages are consumed in the MS-MEETS 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 use PowerShell scripts to remotely call the SharePoint service on the SUT to set up and clean up the test environment of the SUT.

### Adapter abstract layer

Protocol adapter

MS-MEETS adapter interface

There is one property declared in the interface.

Property Url corresponds to the Meetings web service’s entry point URL, which can be set to target the top site or a meeting workspace URL. The test case logic needs to designate the target level of the service entry point URL, so it is abstracted out in the interface for the test case to set.

There are 12 methods declared in the MS-MEETS adapter interface IMS\_MEETSAdapter.

* The 12 methods correspond to the 12 MS-MEETS operations: AddMeeting, AddMeetingFromICal, CreateWorkspace, DeleteWorkspace, GetMeetingsInformation, GetMeetingWorkspaces, RemoveMeeting, RestoreMeeting, SetAttendeeResponse, SetWorkspaceTitle, UpdateMeeting and UpdateMeetingFromICal.
* The operators of the 12 methods are abstracted the same as the operations specified in the MS-MEETS Open Specification.

SUT control adapter

SUT control adapter interface

One method corresponding with the following one function is declared in the SUT control adapter interface IMS\_MEETSSUTControlAdapter.

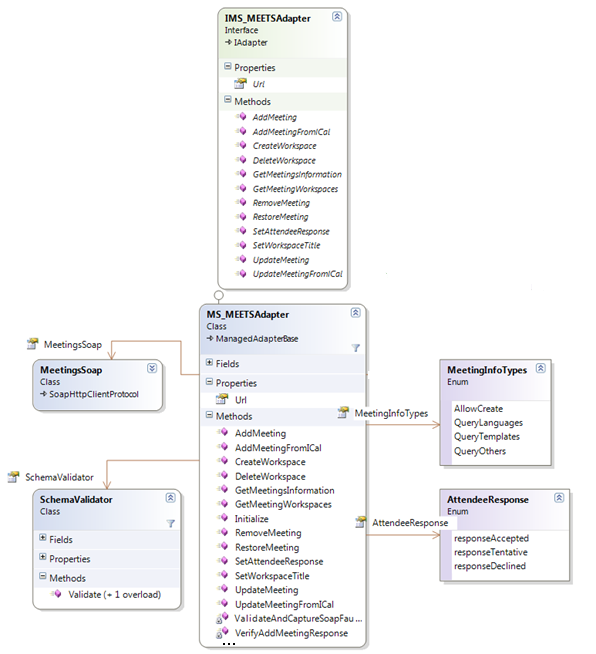
Remove any existing meeting workspaces under the specified site.

### Adapter details

#### Protocol adapter

##### MS-MEETS protocol adapter

The following figure shows the class diagram of the MS-MEETS adapter.



**Protocol adapter class diagram**

The following outlines details of the class diagram:

Adapter interface

* IMS\_MEETSAdapter is the interface of the protocol adapter.
* IMS\_MEETSAdapter defines the 12 protocol operation methods invoked by test cases. See the list of the 12 methods in section [2.4.3 Adapter Abstract Layer.](#_Adapter_abstract_layer)
* IMS\_MEETSAdapter defines one Url property to hold the state of meetings web service’s entry point URL.

Adapter implementation

* MS\_MEETSAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_MEETSAdapter.
* The Initialize method is used to initialize the MS-MEETS test suite.
* The private methods beginning with “Validate” or “Verify” are used to invoke schema validation method and verify requirements related to the message responses. They will be invoked in the protocol operation methods.

Other classes

* SchemaValidator is generated by XMLDOM tool plus the full WSDL in the MS-MEETS. This class is used to validate whether a piece of XML fragment in the response message complies with the XML Schema.
* The MeetingsSoap class is the proxy class generated from the full WSDL whose methods are overridden to support XML validation.

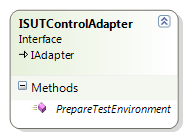
Enumeration

* MeetingInfoTypes is used to specify what information should be returned when calling GetMeetingsInformation.
* AttendeeResponse is used to specify what the attendee response is to the meeting request: accept, decline or tentative.

#### SUT control adapter

##### MS-MEETS 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\_MEETSSUTControlAdapter 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 the in-scope, testable requirements in the MS-MEETS test suite. The details of the scenarios are as follows:

|  |  |
| --- | --- |
| Scenario | Description |
| [S01\_MeetingWorkspace](#_S1_MeetingWorkspace) | This scenario is to add meeting workspace, set workspace’s title, get workspaces information and delete the meeting workspace. |
| [S02\_Meeting](#_S2_Meeting_2) | This scenario is to add meeting, update meeting, delete and restore the meeting. |
| [S03\_MeetingFromICal](#_S3_MeetingFromICal_2) | This scenario is to add, update and meeting to a workspace based on a calendar object. Also include set attendee response. |
| [S04\_RecurringMeeting](#_S4_RecurringMeeting) | This scenario is to add recurring meeting to a workspace. |

MS-MEETS scenario

### S01\_MeetingWorkspace

Description

Adds meeting workspace, sets workspace’s title, gets workspaces information and deletes the meeting workspace.

Operations

* GetMeetingsInformation
* CreateWorkspace
* GetMeetingWorkspaces
* SetWorkspaceTitle
* DeleteWorkspace

Prerequisite

Create a site and remove any existing meeting workspaces in the site.

Cleanup

N/A

### S02\_Meeting

Description

Adds meeting, updates meeting, deletes and restores the meeting.

Operations

* CreateWorkspace
* AddMeeting
* UpdateMeeting
* RemoveMeeting
* RestoreMeeting
* GetMeetingsInformation
* DeleteWorkspace

Prerequisite

Create a site and delete all the possible meeting workspaces in the site.

Cleanup

N/A

### S03\_MeetingFromICal

Description

Adds and updates meeting to a workspace based on a calendar object. Also include sets attendee response.

Operations

* CreateWorkspace
* AddMeetingFromICal
* UpdateMeetingFromICal
* SetAttendeeResponse
* DeleteWorkspace.

Prerequisite

Create a site and delete all the possible meeting workspaces in the site.

Cleanup

N/A

### S04\_RecurringMeeting

Description

Add recurring meeting to a workspace.

Operations

* CreateWorkspace
* AddMeetingFromICal
* GetMeetingsInformation
* GetMeetingWorkspaces
* DeleteWorkspace

Prerequisite

Create a site and delete all the possible meeting workspaces in the site.

Cleanup

N/A

## Test case design

### Traditional test case design

Traditional Testing approach is selected as the test approach for this test suite. 42 test cases are designed to cover the server-side and testable requirements.

**Test case selection**

42 traditional test cases are designed to cover the four scenarios mentioned in section [2.5 Test scenarios](#_Test_Scenarios). Details of the traditional test cases are specified in section [2.6.2 Test case description](#_Test_Cases_Description_1). The scenarios distributions of the test cases are listed in the following table

|  |  |
| --- | --- |
| Scenario ID | Test case name |
| S01\_MeetingWorkspace | [MSMEETS\_S01\_TC01\_WorkspaceOperations](#S1_TC1) |
| [MSMEETS\_S01\_TC02\_CreateWorkspaceOnWorkspaceError](#S1_TC2) |
| [MSMEETS\_S01\_TC03\_WorkspaceInvalidUrlError](#S1_TC3) |
| [MSMEETS\_S01\_TC04\_AllowCreateOnWorkspaceError](#S1_TC4) |
| [MSMEETS\_S01\_TC05\_CreateWorkspaceWithLongTitle](#S1_TC5) |
| [MSMEETS\_S01\_TC06\_SetWorkspaceTitleNotSpecified](#S1_TC6) |
| [MSMEETS\_S01\_TC07\_VerifyUrlOfNewCreatedWorkspace](#S1_TC7) |
| [MSMEETS\_S01\_TC08\_GetMeetingWorkspacesResponse](#S1_TC8) |
| [MSMEETS\_S01\_TC09\_VerifyWorkspaceStatusWhenRequestFlagIs0x8](#S1_TC9) |
| [MSMEETS\_S01\_TC10\_SetWorkspaceTitleWithLongTitle](#S1_TC10) |
| [MSMEETS\_S01\_TC11\_GetMeetingInformationByAllowCreate](#S1_TC11) |
| [MSMEETS\_S01\_TC12\_GetMeetingInformationForSingleMeetingInstance](#S1_TC12) |
| [MSMEETS\_S01\_TC13\_CreateWorkspaceWithAllParametersSpecified](#S1_TC14) |
| [MSMEETS\_S01\_TC14\_CreateWorkspaceWithoutOptionalParameters](#S1_TC15) |
| [MSMEETS\_S01\_TC15\_GetMeetingInformationWithAllParametersSpecified](#S1_TC16) |
| [MSMEETS\_S01\_TC16\_GetMeetingInformationWithoutOptionalParameters](#S1_TC17) |
| S02\_Meeting | [MSMEETS\_S02\_TC01\_MeetingOperations](#S2_TC1) |
| [MSMEETS\_S02\_TC02\_RestoreMeetingError](#S2_TC2) |
| [MSMEETS\_S02\_TC03\_MeetingInvalidUrlError](#S2_TC3) |
| [MSMEETS\_S02\_TC04\_VerifyMeetingCountInWorkspace](#S2_TC4) |
| [MSMEETS\_S02\_TC05\_AddMeetingWithAllParametersSpecified](#S2_TC5) |
| [MSMEETS\_S02\_TC06\_AddMeetingWithoutOptionalParameters](#S2_TC6) |
| [MSMEETS\_S02\_TC07\_UpdateMeetingWithAllParametersSpecified](#S2_TC7) |
| [MSMEETS\_S02\_TC08\_UpdateMeetingWithoutOptionalParameters](#S2_TC8) |
| [MSMEETS\_S02\_TC09\_RemoveMeetingWithAllParametersSpecified](#S2_TC9) |
| [MSMEETS\_S02\_TC10\_RemoveMeetingWithoutOptionalParameters](#S2_TC10) |
| S03\_MeetingFromICal | [MSMEETS\_S03\_TC01\_MeetingFromICalOperations](#S3_TC1) |
| [MSMEETS\_S03\_TC02\_MeetingFromICalInvalidUrlError](#S3_TC2) |
| [MSMEETS\_S03\_TC03\_UpdateMeetingFromICalError](#S3_TC3) |
| [MSMEETS\_S03\_TC04\_AddMeetingFromICalWithInvalidAttendees](#S3_TC4) |
| [MSMEETS\_S03\_TC05\_SetAttendeeResponseForIgnoreUtcDateTimeOrganizerCriticalChange](#S3_TC5) |
| [MSMEETS\_S03\_TC06\_UpdateMeetingFromICalWithInvalidAttendees](#S3_TC6) |
| [MSMEETS\_S03\_TC07\_UpdateMeetingFromICalWithEmptyicalText](#S3_TC9) |
| [MSMEETS\_S03\_TC08\_AddMeetingFromICalWithAllParametersSpecified](#S3_TC10) |
| [MSMEETS\_S03\_TC09\_AddMeetingFromICalWithoutOptionalParameters](#S3_TC11) |
| [MSMEETS\_S03\_TC10\_SetAttendeeResponseWithAllParametersSpecified](#S3_TC12) |
| [MSMEETS\_S03\_TC11\_SetAttendeeResponseWithoutOptionalParameters](#S3_TC13) |
| [MSMEETS\_S03\_TC12\_UpdateMeetingFromICalWithAllParametersSpecified](#S3_TC14) |
| [MSMEETS\_S03\_TC13\_UpdateMeetingFromICalWithoutOptionalParameters](#S3_TC15) |
| [MSMEETS\_S03\_TC14\_UpdateMeetingFromICalWhenICalTextNotPresent](#s03_TC14) |
| S04\_RecurringMeeting | [MSMEETS\_S04\_TC01\_RecurringMeetingOperations](#S4_TC1) |
| [MSMEETS\_S04\_TC02\_RecurringMeetingError](#S4_TC2) |

Test case scenario distribution

The test cases are designed to verify the MS-MEETS response messages and the core operations of this protocol. For example, the request message sent to server is actually verified by the server and the response is sent back to the client with correct result.

### Test case description

Common Prerequisite for all the test cases:

|  |  |
| --- | --- |
| Common prerequisites | Call the SUT control adapter method PrepareTestEnvironment to make sure the test environment is clean. |

Test case common steps

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

The following tables describe the traditional test cases.

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC01\_WorkspaceOperations |
| Description | This test case is used to test typical workspace scenario. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a GetMeetingsInformationSoapIn request message to check if creating workspace is supported and to query available languages.   The protocol server returns a GetMeetingsInformationSoapOut response to indicate whether the server support creating workspace and return the available languages.   1. The protocol client sends a GetMeetingsInformationSoapIn request message to get available workspace templates.   The protocol server returns a GetMeetingsInformationSoapOut response to indicate the available workspace templates.   1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace on the server.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends a GetMeetingWorkspacesSoapIn request to query information about meeting workspaces of the parent web site.   The protocol server returns a GetMeetingWorkspacesSoapOut response about the new created workspace.   1. The protocol client sends a GetMeetingsInformationSoapIn request on the created workspace to get meetings information.   The protocol server returns a GetMeetingsInformationSoapOut response to indicate there should be no meetings on the created workspace.   1. The protocol client sends a SetWorkspaceTitleSoapIn request to set the title of the created meeting workspace.   The protocol server returns a SetWorkspaceTitleSoapOut message to indicate the workspace title has been updated.   1. The protocol client sends a GetMeetingWorkspacesSoapIn request to query information about meeting workspaces of the parent web site.   The protocol server returns a GetMeetingWorkspacesSoapOut response to indicate the workspace title has been updated.   1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the created workspace.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted.   1. The protocol client sends a GetMeetingWorkspacesSoapIn request to query information about meeting workspaces of the parent web site.   The protocol server returns a GetMeetingWorkspacesSoapOut response with no workspace to indicate that the workspace has been successfully deleted in step 8. |
| Cleanup | N/A |

MSMEETS\_S01\_TC01\_WorkspaceOperations

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC02\_CreateWorkspaceOnWorkspaceError |
| Description | This test case verifies that the CreateWorkspace operation cannot create a meeting workspace as a sub site of another meeting workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends another CreateWorkspaceSoapIn request to create another meeting workspace on the meeting workspace created in step 1. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC02\_CreateWorkspaceOnWorkspaceError

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC03\_WorkspaceInvalidUrlError |
| Description | This test case is used to verify the error codes about workspace when the service Url is not a meeting workspace Url. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a SetWorkspaceTitleSoapIn request to specify a name for a web Site that is not a meeting workspace. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete a web site that is not a meeting workspace. |
| Cleanup | N/A |

MSMEETS\_S01\_TC03\_WorkspaceInvalidUrlError

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC04\_AllowCreateOnWorkspaceError |
| Description | This test case verifies that, when the bit flag 0x1 is specified in requestFlags, server returns a SOAP fault if sent the GetMeetingsInformation operation to a web site that is a meeting workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends a GetMeetingsInformationSoapIn request, with the bit flag 0x1 is specified in requestFlags, to query the information about the meeting workspace created in step 1. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC04\_AllowCreateOnWorkspaceError

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC05\_CreateWorkspaceWithLongTitle |
| Description | This test case verifies that if the title in CreateWorkspace is longer than 255 characters, the title will be truncated. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace; the title element in this request is longer than 255 characters.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends a GetMeetingWorkspacesSoapIn request to get meeting workspaces from the site. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC05\_CreateWorkspaceWithLongTitle

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC06\_SetWorkspaceTitleNotSpecified |
| Description | This test case is used to verify the behavior about SetWorkspaceTitle with title absent. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server successfully creates a new meeting workspace on the site.   1. The protocol client sends a SetWorkspaceTitleSoapIn request to update the title of the meeting workspace created in step 1 to null.   The protocol server returns a SetWorkspaceTitleSoapOut response to indicate the workspace title has been update successfully.   1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1. 2. The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted.   The protocol client sends a GetMeetingWorkspacesSoapIn request to get meeting workspaces from the site. |
| Cleanup | N/A |

MSMEETS\_S01\_TC06\_SetWorkspaceTitleNotSpecified

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC07\_VerifyUrlOfNewCreatedWorkspace |
| Description | This test case is used to verify the URL of new created workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call GetMeetingWorkspaces method to get the Url of newly created workspace. 3. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC07\_VerifyUrlOfNewCreatedWorkspace

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC08\_GetMeetingWorkspacesResponse |
| Description | This test case is used to verify the elements in GetMeetingWorkspacesResponse. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call GetMeetingWorkspaces method to get meeting workspace title. 3. Call CreateWorkspace method again to create another workspace on test site. 4. Call GetMeetingWorkspaces method to get meeting workspace length. 5. The protocol client sends a DeleteWorkspaceSoapIn request to delete the 2 meeting workspaces.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the 2 workspaces have been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC08\_GetMeetingWorkspacesResponse

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC09\_VerifyWorkspaceStatusWhenRequestFlagIs0x8 |
| Description | This test case is used to verify the WorkspaceStatus when requestFlags set to 0x8. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call GetMeetingsInformation method to get meeting information on test site.   **Input Parameters:**   * MeetingInfoTypes: QueryOthers  1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC09\_VerifyWorkspaceStatusWhenRequestFlagIs0x8

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC10\_SetWorkspaceTitleWithLongTitle |
| Description | This test case is used to verify the behavior about SetWorkspaceTitle with long title. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site.  2. Call SetWorkspaceTitle method to set workspace title to 255 characters string.  3. Call GetMeetingWorkspaces method to get meeting workspace on test site.  4. Call SetWorkspaceTitle method to set workspace title to 256 characters string.  5. Call GetMeetingWorkspaces method to get meeting workspace on test site again.  6. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.  The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC10\_SetWorkspaceTitleWithLongTitle

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC11\_GetMeetingInformationByAllowCreate |
| Description | This test case is used to verify get meeting information with valid and invalid URL. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site.  2. Call GetMeetingsInformation method with requestFlags bit set to 0x1 to make sure it is the workspace's parent site.  **Input Parameters:**   * MeetingInfoTypes: AllowCreate   3. Call GetMeetingsInformation method with requestFlags bit set to 0x1 to a web site that is a meeting workspace.  4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.  The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC11\_GetMeetingInformationByAllowCreate

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC12\_GetMeetingInformationForSingleMeetingInstance |
| Description | This test case is used to test GetMeetingsInformation operation for single meeting instance. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add two single meetings in workspace. 3. Call GetMeetingsInformation method to get the number of single meeting instances. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC12\_GetMeetingInformationForSingleMeetingInstance

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC13\_CreateWorkspaceWithAllParametersSpecified |
| Description | This test case is used to test create workspace with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call GetMeetingInformation method to check whether creating workspace is supported and query available languages. 2. Call GetMeetingInformation method to get available workspace templates. 3. Call CreateWorkspace method to create a new workspace with all parameters specified. 4. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 3.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC13\_CreateWorkspaceWithAllParametersSpecified

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC14\_CreateWorkspaceWithoutOptionalParameters |
| Description | This test case is used to test create workspace without optional parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a new workspace without optional parameters specified. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S01\_TC14\_CreateWorkspaceWithoutOptionalParameters

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC15\_GetMeetingInformationWithAllParametersSpecified |
| Description | This test case is used to test GetMeetingInformation with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call GetMeetingInformation method to check whether creating workspace is supported and query available languages. 2. Call GetMeetingInformation method with all parameters specified. |
| Cleanup | N/A |

MSMEETS\_S01\_TC15\_GetMeetingInformationWithAllParametersSpecified

|  |  |
| --- | --- |
| S01\_MeetingWorkspace | |
| Test case ID | MSMEETS\_S01\_TC16\_GetMeetingInformationWithoutOptionalParameters |
| Description | This test case is used to test GetMeetingInformation without optional parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call GetMeetingInformation method to GetMeetingInformation without optional parameters specified. |
| Cleanup | N/A |

MSMEETS\_S01\_TC16\_GetMeetingInformationWithoutOptionalParameters

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC01\_MeetingOperations |
| Description | This test case is used to test the typical meeting scenario. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends an AddMeetingSoapIn request to add a new meeting in the meeting workspace created in step 1.   The protocol server returns an AddMeetingSoapOut response to indicate the server adds a new meeting on the workspace successfully.   1. The protocol client sends an UpdateMeetingSoapIn request to update the title and location of the meeting created in step 2.   The protocol server returns an UpdateMeetingSoapOut response to indicate the server updates the meeting on the workspace successfully.   1. The protocol client sends a RemoveMeetingSoapIn request to remove the meeting created in step 2.   The protocol server returns a RemoveMeetingSoapOut response to indicate the server remove the meeting from the workspace successfully.   1. The protocol client sends a RestoreMeetingSoapIn request to restore the meeting that has been removed in step 4.   The protocol server returns a RestoreMeetingSoapOut response to indicate the server restore the meeting successfully.   1. The protocol client sends a GetMeetingsInformationSoapIn request to query the information about the meeting on the workspace. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC01\_MeetingOperations

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC02\_RestoreMeetingError |
| Description | This test case is used to verify the error when restoring a non-existent meeting. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends a RestoreMeetingSoapIn request to restore a meeting that does not exist in the meeting workspace created in step 1. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC02\_RestoreMeetingError

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC03\_MeetingInvalidUrlError |
| Description | This test case verifies that server returns a SOAP fault when the AddMeeting and UpdateMeeting operations were sent to a web site that is not a meeting workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends an AddMeetingSoapIn request to a web site that is not a meeting workspace. 2. The protocol client sends an UpdateMeetingSoapIn request to a web site that is not a meeting workspace. |
| Cleanup | N/A |

MSMEETS\_S02\_TC03\_MeetingInvalidUrlError

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC04\_VerifyMeetingCountInWorkspace |
| Description | This test case is used to verify the meeting count under workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add one meeting on new created workspace. 3. Call GetMeetingsInformation method to get workspace status, make sure there is only one meeting in workspace. 4. Call AddMeeting method again to add another meeting on workspace. 5. Call GetMeetingsInformation method to get workspace status, make sure there are two meetings in workspace. 6. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC04\_VerifyMeetingCountInWorkspace

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC05\_AddMeetingWithAllParametersSpecified |
| Description | This test case is used to test add meeting with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting with all parameters specified. 3. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC05\_AddMeetingWithAllParametersSpecified

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC06\_AddMeetingWithoutOptionalParameters |
| Description | This test case is used to test add meeting without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting without optional parameters. 3. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC06\_AddMeetingWithoutOptionalParameters

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC07\_UpdateMeetingWithAllParametersSpecified |
| Description | This test case is used to test update meeting with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting in the workspace. 3. Call UpdateMeeting method to update the meeting with all parameters specified. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC07\_UpdateMeetingWithAllParametersSpecified

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC08\_UpdateMeetingWithoutOptionalParameters |
| Description | This test case is used to test update meeting without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting in the workspace. 3. Call UpdateMeeting method to update the meeting without optional parameters. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC08\_UpdateMeetingWithoutOptionalParameters

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC09\_RemoveMeetingWithAllParametersSpecified |
| Description | This test case is used to test remove meeting with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting in the workspace. 3. Call RemoveMeeting method to remove the meeting with all parameters specified. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC09\_RemoveMeetingWithAllParametersSpecified

|  |  |
| --- | --- |
| S02\_Meeting | |
| Test case ID | MSMEETS\_S02\_TC10\_RemoveMeetingWithoutOptionalParameters |
| Description | This test case is used to test remove meeting without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeeting method to add a meeting in the workspace. 3. Call RemoveMeeting method to remove the meeting without optional parameters. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S02\_TC10\_RemoveMeetingWithoutOptionalParameters

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC01\_MeetingFromICalOperations |
| Description | This test case is used to test typical meeting based on calendar object scenario. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends an AddMeetingFromICalSoapIn request to add a new meeting in the meeting workspace created in step 1.   The protocol server returns an AddMeetingFromICalSoapOut response to indicate the server adds a meeting based on calendar object on the workspace successfully.   1. The protocol client sends an UpdateMeetingFromICalSoapIn request to update the meeting created in step 2. 2. The protocol client sends a SetAttendeeResponseSoapIn request to specify attendee response to the meeting added in step 2.   The protocol server returns a SetAttendeeResponseSoapOut response to indicate the server set the response of the attendee of the meeting successfully.   1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC01\_MeetingFromICalOperations

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC02\_MeetingFromICalInvalidUrlError |
| Description | This test case is used to verify the error when the operations AddMeetingFromICal, UpdateMeetingFromICal and SetAttendeeResponse are sent to a web site that is not a meeting workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends an AddMeetingFromICalSoapIn request to a web site that is not a meeting workspace. 2. The protocol client sends an UpdateMeetingFromICalSoapIn request to a web site that is not a meeting workspace. 3. The protocol client sends a SetAttendeeResponseSoapIn request to a web site that is not a meeting workspace. |
| Cleanup | N/A |

MSMEETS\_S03\_TC02\_MeetingFromICalInvalidUrlError

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC03\_UpdateMeetingFromICalError |
| Description | This test case is used to verify the error of UpdateMeeting with invalid parameter. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server creates a new meeting workspace on the site successfully.   1. The protocol client sends an AddMeetingFromICalSoapIn request to add a new meeting in the meeting workspace created in step 1.   The protocol server returns an AddMeetingFromICalSoapOut response to indicate the server adds a meeting based on calendar object on the workspace successfully.   1. The protocol client sends an UpdateMeetingFromICalSoapIn request to update the meeting added in step 2; the icalText element in this request is empty. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC03\_UpdateMeetingFromICalError

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC04\_AddMeetingFromICalWithInvalidAttendees |
| Description | This test case is used to test AddMeetingFromICal operation when the parameter icalText contains more than 254 attendees' elements. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call GetICalendar method to get icalendar with invalid attendees.   **Input Parameters:**   * Icalendar: 255  1. Call AddMeetingFromICal method to add a meeting from ICalendar with invalid attendees. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC04\_AddMeetingFromICalWithInvalidAttendees

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC05\_SetAttendeeResponseForIgnoreUtcDateTimeOrganizerCriticalChange |
| Description | This test case is used to test SetAttendeeResponse operation when server ignores utcDateTimeOrganizerCriticalChange element and uses only utcDateTimeAttendeeCriticalChange. |
| Prerequisites | The product should be Windows SharePoint Services 3.0 |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to add a meeting from ICalendar. 3. Call SetAttendeeResponse method to set attendee response.   **Input Parameters:**   * utcDateTimeOrganizerCriticalChange: DateTime.Now.AddHours(1) * utcDateTimeAttendeeCriticalChange: DateTime.Now.AddHours(2)  1. Call SetAttendeeResponse method to set attendee response again.   **Input Parameters:**   * utcDateTimeOrganizerCriticalChange: DateTime.Now.AddHours(2) * utcDateTimeAttendeeCriticalChange: DateTime.Now.AddHours(2)  1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC05\_SetAttendeeResponseForIgnoreUtcDateTimeOrganizerCriticalChange

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC06\_UpdateMeetingFromICalWithInvalidAttendees |
| Description | This test case is used to test UpdateMeetingFromICal operation when the parameter icalText contains more than 254 attendees' elements. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to add a meeting. 3. Call UpdateMeetingFromICal method to update meeting from ICalendar with more than 254 attendees. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC06\_UpdateMeetingFromICalWithInvalidAttendees

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC07\_UpdateMeetingFromICalWithEmptyicalText |
| Description | This test case is used to test UpdateMeetingFromICal operation when the parameter icalText is empty. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call UpdateMeetingFromICal method to update meeting from ICalendar with empty icalText.   **Input Parameters:**   * icalText: null  1. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC07\_UpdateMeetingFromICalWithEmptyicalText

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC08\_AddMeetingFromICalWithAllParametersSpecified |
| Description | This test case is used to test add meeting from icalendar with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar with all parameters specified. 3. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC08\_AddMeetingFromICalWithAllParametersSpecified

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC09\_AddMeetingFromICalWithoutOptionalParameters |
| Description | This test case is used to test add meeting from icalendar without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar without optional parameters. 3. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC09\_AddMeetingFromICalWithoutOptionalParameters

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC10\_SetAttendeeResponseWithAllParametersSpecified |
| Description | This test case is used to test SetAttendeeResponse operation with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call SetAttendeeResponse method to set attendee response with all parameters specified. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC10\_SetAttendeeResponseWithAllParametersSpecified

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC11\_SetAttendeeResponseWithoutOptionalParameters |
| Description | This test case is used to test SetAttendeeResponse operation without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call SetAttendeeResponse method to set attendee response without optional parameters. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC11\_SetAttendeeResponseWithoutOptionalParameters

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC12\_UpdateMeetingFromICalWithAllParametersSpecified |
| Description | This test case is used to test UpdateMeetingFromICal operation with all parameters specified. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call UpdateMeetingFromICal method to update the meeting with all parameters specified. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC12\_UpdateMeetingFromICalWithAllParametersSpecified

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC13\_UpdateMeetingFromICalWithoutOptionalParameters |
| Description | This test case is used to test UpdateMeetingFromICal operation without optional parameters. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call UpdateMeetingFromICal method to update the meeting without optional parameters. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC13\_UpdateMeetingFromICalWithoutOptionalParameters

|  |  |
| --- | --- |
| S03\_MeetingFromICal | |
| Test case ID | MSMEETS\_S03\_TC14\_UpdateMeetingFromICalWhenICalTextNotPresent |
| Description | This test case is used to test UpdateMeetingFromICal operation when icalText is not present. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. Call CreateWorkspace method to create a workspace on test site. 2. Call AddMeetingFromICal method to get meeting from ICalendar. 3. Call UpdateMeetingFromICal method to update the meeting without icalText parameter. 4. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S03\_TC14\_UpdateMeetingFromICalWhenICalTextNotPresent

|  |  |
| --- | --- |
| S04\_RecurringMeeting | |
| Test case ID | MSMEETS\_S04\_TC01\_RecurringMeetingOperations |
| Description | This test case is used to verify the recurring meeting related requirements. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends 3 CreateWorkspaceSoapIn requests to create the 3 meeting workspaces on the site.   The protocol server returns 3 CreateWorkspaceSoapOut responses to indicate the server creates 3 new meeting workspaces on the site successfully separately.   1. The protocol client sends an AddMeetingSoapIn request message to add a single instance meeting in the second meeting workspace created in step 1.   The protocol server returns an AddMeetingSoapOut response to indicate the server adds a new meeting on the workspace successfully.   1. The protocol client sends an AddMeetingFromICalSoapIn request message to add a recurring meeting in the third meeting workspace created in step 1.   The protocol server returns an AddMeetingFromICalSoapOut response to indicate the server adds a recurring meeting on the workspace successfully.   1. The protocol client sends a GetMeetingsInformationSoapIn request message to the first meeting workspace to query the meeting information on the workspace.   The protocol server returns a GetMeetingsInformationSoapOut response to indicate that the first meeting workspace does not contain any meeting instance.   1. The protocol client sends a GetMeetingsInformationSoapIn request message to the second meeting workspace to query the meeting information on the workspace.   The protocol server returns a GetMeetingsInformationSoapOut response to indicate that the second meeting workspace contains single instance meeting.   1. The protocol client sends a GetMeetingsInformationSoapIn request message to the third meeting workspace to query the meeting information on the workspace. 2. The protocol client sends GetMeetingWorkspaces to get available meeting workspaces for non-recurring meeting. 3. The protocol client sends GetMeetingWorkspaces to get available meeting workspaces for recurring meeting. 4. The protocol client sends 3 DeleteWorkspaceSoapIn requests to delete the meeting workspaces created in step 1.   The protocol server returns 3 DeleteWorkspaceSoapOut responses to indicate the workspaces have been deleted. |
| Cleanup | N/A |

MSMEETS\_S04\_TC01\_RecurringMeetingOperations

|  |  |
| --- | --- |
| S04\_RecurringMeeting | |
| Test case ID | MSMEETS\_S04\_TC02\_RecurringMeetingError |
| Description | This test case is used to verify the error code when adding recurring meeting to a non-empty workspace. |
| Prerequisites | Common prerequisites |
| Test execution steps | 1. The protocol client sends a CreateWorkspaceSoapIn request to create a new meeting workspace.   The protocol server returns a CreateWorkspaceSoapOut response to indicate the server successfully creates a new meeting workspace on the site.   1. The protocol client sends an AddMeetingSoapIn request message to add a single instance meeting in the meeting workspace created in step 1.   The protocol server returns an AddMeetingSoapOut response to indicate the server adds a new meeting on the workspace successfully.   1. The protocol client sends an AddMeetingFromICalSoapIn request message to add a recurring meeting in the meeting workspace created in step 1. 2. The protocol client sends a DeleteWorkspaceSoapIn request to delete the meeting workspace created in step 1.   The protocol server returns a DeleteWorkspaceSoapOut response to indicate the workspace has been deleted. |
| Cleanup | N/A |

MSMEETS\_S04\_TC02\_RecurringMeetingError