

MS-SHDACCWS Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how MS-SHDACCWS test suite is designed to test MS-SHDACCWS 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](#_Toc402370090)

[1.1 Configuring the test suite client 3](#_Toc402370091)

[1.1.1 Configuring the test suite client manually 3](#_Toc402370092)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc402370093)

[1.2 Configuring the system under test (SUT) 4](#_Toc402370094)

[1.2.1 Configuring the SUT manually 4](#_Toc402370095)

[1.2.2 Configuring the SUT by scripts 4](#_Toc402370096)

[1.3 Configuring the SHOULD/MAY requirements 4](#_Toc402370097)

[2 Test suite design 5](#_Toc402370098)

[2.1 Assumptions, scope and constraints 5](#_Toc402370099)

[2.2 Test suite architecture 5](#_Toc402370100)

[2.3 Technical dependencies and considerations 7](#_Toc402370101)

[2.4 Adapter design 7](#_Toc402370102)

[2.4.1 Adapter overview 7](#_Toc402370103)

[2.4.2 Technical feasibility of adapter approach 8](#_Toc402370104)

[2.4.3 Adapter abstract layer 8](#_Toc402370105)

[2.4.4 Adapter details 9](#_Toc402370106)

[2.5 Test scenarios 11](#_Toc402370107)

[2.5.1 S01\_ VerifyIsSingleClient 11](#_Toc402370108)

[2.6 Test case design 12](#_Toc402370109)

[2.6.1 Traditional test case design 12](#_Toc402370110)

[2.6.2 Test case description 12](#_Toc402370111)

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

1. Open MS-SHDACCWS\TestSuite\MS-SHDACCWS\_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 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 the SUT settings and configuration.

* Property name="SiteCollectionName" value="MS-SHDACCWS\_SiteCollection"
* Property name="FileIdOfLock" value="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" The file id of a file uploaded in SUT, which is applied an exclusive lock.The id is the unique identifier of the file in the SUT
* Property name="FileIdOfCoAuthoring" value="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" The file id of a file uploaded in SUT, which is applied a Co-Authoring request. The id is the unique identifier of the file in the SUT.
* Property name="FileIdOfNormal" value="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" The file id of a file uploaded in SUT without any other locks or Co-Authoring status. The id is the unique identifier of the file in the SUT.
* Property name="TargetServiceUrl" value="[TransportType]://[SUTComputerName]/sites/[SiteCollectionName]/\_vti\_bin/sharedaccess.asmx"

1. The following properties are not associated with SUT settings and can normally retain the default values.

* Property name="ServiceTimeOut" value="10"

### 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

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-SHDACCWS\_WindowsSharePointServices3\_SHOULDMAY.deployment.ptfconfig
* MS-SHDACCWS\_SharePointServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-SHDACCWS\_SharePointFoundation2010\_SHOULDMAY.deployment.ptfconfig
* MS-SHDACCWS\_SharePointServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-SHDACCWS\_SharePointFoundation2013\_SHOULDMAY.deployment.ptfconfig
* MS-SHDACCWS\_SharePointServer2013\_SHOULDMAY.deployment.ptfconfig

If RXXXEnabled is set to true, the requirement must be run. If false, the requirement must not be run. 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-SHDACCWS\_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 the operation: IsOnlyClient.
* 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 HTTP, HTTPS, SOAP1.1 and SOAP1.2.

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 Open Specification. The following figure shows the architecture of this test suite.



The architecture of the test suite

The details of the MS-SHDACCWS test suite architecture

* SUT hosts the Shared Access Web Service which this test suite runs against.
* From a third-party’s point of view, the SUT is the protocol server implementation.
* The following products have been tested with the MS-SHDACCWS test suite on the Windows platform.
* Microsoft SharePoint Foundation 2010 SP2
* Microsoft SharePoint Foundation 2013 SP1
* Microsoft SharePoint Server 2010 SP2
* Microsoft SharePoint Server 2013 SP1
* The test suite acts as the client to communicate with the SUT and validates the requirements gathered from the MS-SHDACCWS Open Specification.
* Test cases use the MS-SHDACCWS adapter to call and get the results of the MS-SHDACCWS operations.
* MS-SHDACCWS adapter is used in the test cases. The test cases call the method in the interface to invoke the one protocol adapter 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 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 .NET Framework SDK to generate the MS-SHDACCWS proxy class.
* This test suite depends on Protocol Test Framework (PTF) to derive managed adapters.

Encryption consideration

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

## Adapter design

### Adapter overview

One protocol adapter and one SUT control adapter will be designed for this test suite.

Protocol adapter

There is 1 protocol adapter: MS-SHDACCWS adapter. The MS-SHDACCWS protocol adapter is managed adapter, which is derived from the ManagedAdapterBase class in PTF.

* MS-SHDACCWS adapter
* Chooses HTTP or HTTPS and SOAP 1.1 or SOAP 1.2 for transport;
* Constructs requests of two MS-SHDACCWS operations;
* Communicates with the SUT by sending requests to the SUT and receives the corresponding responses from the SUT;
* Parses the responses messages and validates the messages according to the WSDL schema;
* Generates the result log.
* The MS-SHDACCWS 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 adapters

* The MS-SHDACCWS SUT control adapter uses PowerShell script implementation.
* The MS-SHDACCWS SUT control adapter has the following functionalities
* Set the Co-authoring status for the specified file which is identified by the property "FileIdOfCoAuthoring".
* Set specified status of exclusive lock to the specified file which is identified by the property "FileIdOfLock".
* The SUT control adapter is invoked by the test cases.

### Technical feasibility of adapter approach

Message generation

The MS-SHDACCWS adapter gets the parameter values of the WSDL operations and calls the corresponding operations in MS-SHDACCWS proxy class, the MS-SHDACCWS 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-SHDACCWS proxy class and be passed upon to the MS-SHDACCWS adapter. Then these messages are consumed in the MS-SHDACCWS adapter to validate the message format and to validate the logic-related requirements in the test cases.

SUT control adapter

The SUT control adapter is designed to remotely control the SUT to set the co-authoring status for a file in the server, or lock a file in the server.

### Adapter abstract layer

Protocol adapters

MS-SHDACCWS adapter interface

There is one method declared in the MS-SHDACCWS adapter interface IMS\_SHDACCWSAdapter. The operator of this method is abstracted from the operation specified in the MS-SHDACCWS open specifiation.

The method is described in the following table:

|  |  |  |
| --- | --- | --- |
| No. | Methods | Description |
| 1 | IsOnlyClient | Determine a file whether is in Co-authoring status. |

MS-SHDACCWS adapter interface methods

SUT control adapter

* MS-SHDACCWS SUT control adapter interface
* There are two methods declared in the SUT control adapter interface IMS\_SHDACCWSSUTControlAdapter.

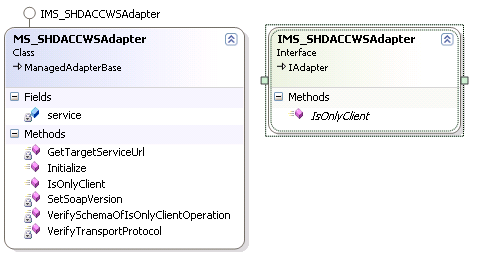
|  |  |  |
| --- | --- | --- |
| No. | Methods | Description |
| 1 | SUTSetCoAuthoringStatus | Set the Co-authoring status for the specified file under the specified Document LibraryName list. The specified file is identified by the property "FileIdOfCoAuthoring". |
| 2 | SUTSetExclusiveLock | Set status of exclusive lock to the specified file which is identified by the property "FileIdOfLock". |

### Adapter details

#### Protocol adapter

##### MS-SHDACCWS protocol adapter

The following figure illustrates the MS-SHDACCWS protocol adapter class diagram.



MS-SHDACCWS adapter class diagram

The following outlines details of the class diagram:

Adapter interface

* The IMS\_SHDACCWS is the interface of MS\_SHDACCWSAdapter.
* The IMS\_SHDACCWS defines the methods invoked by test cases.

Adapter implementation

* MS-SHDACCWSAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_SHDACCWSAdapter.
* The IsOnlyClient method is implemented by generating SOAP request, and then invoking this method provided by the MS-SHDACCWS proxy class to send SOAP request; getting corresponding de-serialized response and verifying schema definition.
* 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.
* The Initialize method is used to initialize the MS-SHDACCWS protocol adapter.

#### SUT control adapter

##### MS-SHDACCWS 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:

IMS\_SHDACCWSSUTControlAdapter is interface of SUT control adapter class which defined all the methods in SUT control adapter class.

## Test scenarios

One scenario is designed to cover the server-side, testable requirements in the MS-SHDACCWS test suite. The details of the scenario is as follow

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_VerifyIsSingleClient | This scenario is used to judge whether it is the only client currently editing a document stored on a collaboration server, or alternately, whether should transition to a shared editing mode. |

MS-SHDACCWS scenario

### S01\_ VerifyIsSingleClient

Description

This scenario is used to judge whether it is the only client currently editing a document stored on a collaboration server, or alternately, whether should transition to a shared editing mode.

Operations

* IsOnlyClient

Prerequisites

Create/upload one document in an existing document library on the server.

Cleanup

None

## 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 4 traditional test cases designed to cover the three scenarios mentioned in section [2.6 Test scenarios.](#OLE_LINK6) Details of the traditional test cases are specified in section [2.7.2 Test case description](#OLE_LINK5). The scenarios distributions of the test cases are listed in the following table.

|  |  |
| --- | --- |
| Scenario ID | Test case name |
| S01\_VerifyIsSingleClient | MSSHDACCWS\_S01\_TC01\_CoAuthoringIsAsked |
| MSSHDACCWS\_S01\_TC02\_NoClientAuthoring |
| MSSHDACCWS\_S01\_TC03\_OnlyOneClientAuthoring |
| MSSHDACCWS\_S01\_TC04\_FileNotExistOnServer |

Test case scenario distribution

### Test case description

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

The following tables describe the traditional test cases.

|  |  |
| --- | --- |
| **S01\_VerifyIsSingleClient** | |
| **Test case ID** | MSSHDACCWS\_S01\_TC01\_CoAuthoringIsAsked |
| **Description** | Verify that the client can get IsOnlyClientSoapOut messages for IsOnlyClient operation, and the server returns "false" when there was a co-authoring transition request for the document. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set the Co-authoring status for the specified file which is identified by the property "FileIdOfCoAuthoring". 2. Get an identifier of the document which there was a co-authoring transition request for. 3. Call method IsOnlyClient with the identifier of the document which there was a co-authoring transition request for. |
| **Cleanup** | N/A |

MSSHDACCWS\_ S01\_TC01\_CoAuthoringIsAsked

|  |  |
| --- | --- |
| **S01\_VerifyIsSingleClient** | |
| **Test case ID** | MSSHDACCWS\_S01\_TC02\_NoClientAuthoring |
| **Description** | Verify that the client can get IsOnlyClientSoapOut messages for IsOnlyClient operation, and the server returns "true" when no client is editing the document. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Get an identifier of the document that no client is editing it. 2. Call method IsOnlyClient with the identifier of the document that no client is editing it. |
| **Cleanup** | N/A |

MSSHDACCWS\_ S01\_TC02\_NoClientAuthoring

|  |  |
| --- | --- |
| **S01\_VerifyIsSingleClient** | |
| **Test case ID** | MSSHDACCWS\_S01\_TC03\_OnlyOneClientAuthoring |
| **Description** | Verify that the client can get IsOnlyClientSoapOut messages for IsOnlyClient operation, and the server returns "true" when the document is currently edited by one client. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set specified status of exclusive lock to the specified file which is identified by the property "FileIdOfLock". 2. Get an identifier of the document that is currently edited by one client. 3. Call method IsOnlyClient with the identifier of the document that is currently edited by one client. |
| **Cleanup** | N/A |

MSSHDACCWS\_S01\_TC03\_OnlyOneClientAuthorin

|  |  |
| --- | --- |
| **S01\_VerifyIsSingleClient** | |
| **Test case ID** | MSSHDACCWS\_S01\_TC04\_FileNotExistOnServer |
| **Description** | Verify that the client can get IsOnlyClientSoapOut messages for IsOnlyClient operation, and the server returns "true" when the document specified by the id can't be found on the server. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call method IsOnlyClient with the identifier of the document that specified by the id can't be found on the server. |
| **Cleanup** | N/A |

MSSHDACCWS\_S01\_TC04\_FileNotExistOnServe

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

|  |  |
| --- | --- |
| Common prerequisites | None |
| Common cleanup | None |

Test case common steps