

MS-AUTHWS Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how MS-AUTHWS test suite is designed to test MS-AUTHWS 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](#_Toc387068986)

[1.1 Configuring the test suite client 3](#_Toc387068987)

[1.1.1 Configuring the test suite client manually 3](#_Toc387068988)

[1.1.2 Configuring the test suite client by scripts 4](#_Toc387068989)

[1.2 Configuring the system under test (SUT) 4](#_Toc387068990)

[1.2.1 Configuring the SUT manually 4](#_Toc387068991)

[1.2.2 Configuring the SUT by scripts 4](#_Toc387068992)

[1.3 Configuring the SHOULD/MAY requirements 4](#_Toc387068993)

[2 Test suite design 5](#_Toc387068994)

[2.1 Assumptions, scope and constraints 5](#_Toc387068995)

[2.2 Test suite architecture 5](#_Toc387068996)

[2.3 Technical dependencies and considerations 6](#_Toc387068997)

[2.4 Adapter design 7](#_Toc387068998)

[2.4.1 Adapter overview 7](#_Toc387068999)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387069000)

[2.4.3 Adapter abstract layer 8](#_Toc387069001)

[2.4.4 Adapter details 8](#_Toc387069002)

[2.5 Test scenarios 10](#_Toc387069003)

[2.5.1 S01\_LoginApplicationUnderFormsAuthentication 10](#_Toc387069004)

[2.5.2 S02\_LoginApplicationUnderNoneAuthentication 10](#_Toc387069005)

[2.5.3 S03\_LoginApplicationUnderWindowsAuthentication 11](#_Toc387069006)

[2.5.4 S04\_LoginApplicationUnderPassportAuthentication 11](#_Toc387069007)

[2.6 Test case design 12](#_Toc387069008)

[2.6.1 Traditional test case design 12](#_Toc387069009)

[2.6.2 Test case description 12](#_Toc387069010)

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

1. Open MS-AUTHWS\TestSuite\MS-AUTHWS\_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 property values to match SUT settings and configuration.

* Property name="WebServiceRelativeUrl" value="/\_vti\_bin/authentication.asmx"
* Property name="WindowsAuthenticationPortForHTTP" value="80"
* Property name="WindowsAuthenticationUrlForHTTP" value="[TransportType]://[SutComputerName]:[WindowsAuthenticationPortForHTTP][WebServiceRelativeUrl]"
* Property name="WindowsAuthenticationPortForHTTPS" value="443"
* Property name="WindowsAuthenticationUrlForHTTPS" value="[TransportType]://[SutComputerName]:[WindowsAuthenticationPortForHTTPS][WebServiceRelativeUrl]"
* Property name="FormsAuthenticationPortForHTTP" value="90"
* Property name="FormsAuthenticationUrlForHTTP" value="[TransportType]://[SutComputerName]:[FormsAuthenticationPortForHTTP][WebServiceRelativeUrl]"
* Property name="FormsAuthenticationPortForHTTPS" value="93"
* Property name="FormsAuthenticationUrlForHTTPS" value="[TransportType]://[SutComputerName]:[FormsAuthenticationPortForHTTPS][WebServiceRelativeUrl]"
* Property name="NoneAuthenticationPortForHTTP" value="92"
* Property name="NoneAuthenticationUrlForHTTP" value="[TransportType]://[SutComputerName]:[NoneAuthenticationPortForHTTP][WebServiceRelativeUrl]"
* Property name="NoneAuthenticationPortForHTTPS" value="95"
* Property name="NoneAuthenticationUrlForHTTPS" value="[TransportType]://[SutComputerName]:[NoneAuthenticationPortForHTTPS][WebServiceRelativeUrl]"
* Property name="PassportAuthenticationPortForHTTP" value="91"
* Property name="PassportAuthenticationUrlForHTTP" value="[TransportType]://[SutComputerName]:[PassportAuthenticationPortForHTTP][WebServiceRelativeUrl]"
* Property name="PassportAuthenticationPortForHTTPS" value="94"
* Property name="PassportAuthenticationUrlForHTTPS" value="[TransportType]://[SutComputerName]:[PassportAuthenticationPortForHTTPS][WebServiceRelativeUrl]"

### 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-AUTHWS\_WindowsSharePointServices3\_SHOULDMAY.deployment.ptfconfig
* MS-AUTHWS\_SharePointServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-AUTHWS\_SharePointFoundation2010\_SHOULDMAY.deployment.ptfconfig
* MS-AUTHWS\_SharePointServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-AUTHWS\_SharePointFoundation2013\_SHOULDMAY.deployment.ptfconfig
* MS-AUTHWS\_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-AUTHWS\_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 three operations: Login, Mode and SwitchWebApplication.
* 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 about login success with Windows or Passport authentication mode, because this operation just support forms authentication mode.
* 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-AUTHWS test suite architecture

* SUT hosts the Authentication 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-AUTHWS test suite on the Windows platform.
* Windows SharePoint Services 3.0 Service Pack 3 (SP3)
* Microsoft SharePoint Foundation 2010 Service Pack 1 (SP2)
* Microsoft SharePoint Foundation 2013 SP1
* Microsoft Office SharePoint Server 2007 Service Pack 3 (SP3)
* Microsoft SharePoint Server 2010 Service Pack 1 (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-AUTHWS Open Specification.
* Test cases use the MS-AUTHWS adapter to call and get the results of the MS-AUTHWS operations.
* MS-AUTHWS adapter is used in the test cases. The test cases call the methods in the interfaces to invoke the protocol adapters’ operations.

## 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-AUTHWS proxy class.
* This test suite depends on Protocol Test Framework (PTF) to derive managed adapters.

Encryption consideration

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

## Adapter design

### Adapter overview

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

Protocol adapter

* MS-AUTHWS protocol adapter
* The MS-AUTHWS adapter is a managed adapter, which is derived from the ManagedAdapterBase class in PTF.
* The MS-AUTHWS adapter has the following functionalities
* Choose HTTP or HTTPS and SOAP 1.1 or 1.2 for transport;
* Construct requests of two MS-AUTHWS operations;
* Communicate with the SUT by sending requests to the SUT and receive the corresponding responses from the SUT;
* Analyze the grammar of the response messages and validate the messages according to the WSDL schema;
* The MS-AUTHWS 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

* None

### Technical feasibility of adapter approach

Message generation

* The MS-AUTHWS adapter gets the parameter values of the WSDL operations and calls the corresponding operations in MS-AUTHWS proxy class, the MS-AUTHWS 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-AUTHWS proxy class.

Message consumption

* The messages received from the SUT will be analyzed for grammar in the MS-AUTHWS proxy class and be passed upon to the MS-AUTHWS adapter. Then these messages are consumed in the MS-AUTHWS adapter to validate the message format and to validate the logic-related requirements in the test cases.

SUT control adapter

* None

### Adapter abstract layer

Protocol adapters

MS-AUTHWS adapter interface

There are three methods declared in the MS-AUTHWS adapter interface IMS\_AUTHWSAdapter.

The methods are described in the following table:

|  |  |  |
| --- | --- | --- |
| No. | Methods | Description |
| 1 | Login | Logs a user onto a Web application (1) by using the user’s logon name and password. |
| 2 | Mode | Retrieves the authentication mode that a Web application (1) uses. |
| 3 | SwitchWebApplication | Switch to the corresponding WebApplication according to AuthenticationMode. |

MS-AUTHWS adapter interface methods

SUT control adapter

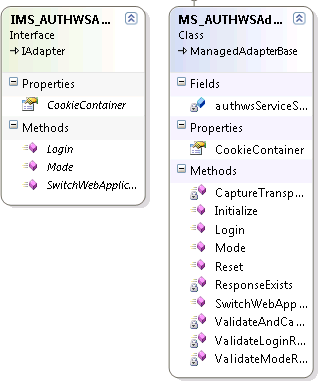
* None

### Adapter details

#### Protocol adapter

##### MS-AUTHWS protocol adapter

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



MS-AUTHWS Protocol adapter class diagram

The following outlines details of the class diagram:

Adapter interface

* The IMS\_AUTHWSAdapter is the interface of MS\_AUTHWSAdapter.
* The IMS\_AUTHWSAdapter defines the methods invoked by test cases, including Login, Mode and SwitchWebApplication.

Adapter implementation

* MS\_AUTHWSAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_ AUTHWSAdapter.
* MS\_AUTHWSAdapter invoking the Login and Mode provided by the MS-AUTHWS proxy class to send message requests, and then getting corresponding de-serialized response and verifying related requirements.
* The Initialize method is used to initialize the MS-AUTHWS protocol adapter.

#### SUT control adapter

* None

## Test scenarios

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_LoginApplicationUnderFormsAuthentication | This scenario is designed to test login application under Forms authentication mode. |
| S02\_LoginApplicationUnderNoneAuthentication | This scenario is designed to test login application under None authentication mode. |
| S03\_LoginApplicationUnderWindowsAuthentication | This scenario is designed to test login application under Windows authentication mode. |
| S04\_LoginApplicationUnderPassportAuthentication | This scenario is designed to test login application under Passport authentication mode. |

MS-AUTHWS scenario

### S01\_LoginApplicationUnderFormsAuthentication

Description

This scenario is designed to test login application under Forms authentication mode.

Operations

* Mode operation
* Login operation

Prerequisites

Create a web application for Forms in IIS.

Cleanup

N/A

### S02\_LoginApplicationUnderNoneAuthentication

Description

This scenario is designed to test login application under None authentication mode.

Operations

* Mode operation
* Login operation

Prerequisites

Create a web application for None in IIS.

Cleanup

N/A

### S03\_LoginApplicationUnderWindowsAuthentication

Description

This scenario is designed to test login application under Windows authentication mode.

Operations

* Mode operation
* Login operation

Prerequisites

N/A

Cleanup

N/A

### S04\_LoginApplicationUnderPassportAuthentication

Description

This scenario is designed to test login application under Passport authentication mode.

Operations

* Mode operation
* Login operation

Prerequisites

Create a web application for Passport in IIS.

Cleanup

N/A

## Test case design

### Traditional test case design

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

Six 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 scenario distributions of the test cases are listed in the following table.

|  |  |
| --- | --- |
| **Scenario ID** | **Test case name** |
| S01\_LoginApplicationUnderFormsAuthentication | MSAUTHWS\_S01\_TC01\_VerifyLoginUnderFormsAuthentication |
| MSAUTHWS\_S01\_TC02\_VerifyLoginUnderFormsAuthenticationWithInvalidUserName |
| MSAUTHWS\_S01\_TC03\_VerifyLoginUnderFormsAuthenticationWithInvalidPassword |
| S02\_LoginApplicationUnderNoneAuthentication | MSAUTHWS\_S02\_TC01\_VerifyLoginUnderNoneAuthentication |
| S03\_LoginApplicationUnderWindowsAuthentication | MSAUTHWS\_S03\_TC01\_VerifyLoginUnderWindowsAuthentication |
| S04\_LoginApplicationUnderPassportAuthentication | MSAUTHWS\_S04\_TC01\_VerifyLoginUnderPassportAuthentication |

Test case scenario distribution

The test cases are designed to verify the MS-AUTHWS 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 | None. |

The following tables describe the traditional test cases.

|  |  |
| --- | --- |
| **S01\_LoginApplicationUnderFormsAuthentication** | |
| **Test case ID** | MSAUTHWS\_S01\_TC01\_VerifyLoginUnderFormsAuthentication |
| **Description** | This test case is used to verify the Login operation under Forms authentication mode should be successful. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode Forms. 2. The client calls the Login method with valid user name and password to login the application with Forms mode. 3. Verify the returned cookie name and value in Login result. |
| **Cleanup** | N/A |

MSAUTHWS\_S01\_TC01\_ VerifyLoginUnderFormsAuthentication

|  |  |
| --- | --- |
| **S01\_LoginApplicationUnderFormsAuthentication** | |
| **Test case ID** | MSAUTHWS\_S01\_TC02\_VerifyLoginUnderFormsAuthenticationWithInvalidUserName |
| **Description** | This test case is used to verify the Login operation under Forms authentication mode is failed with invalid user name. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode Forms. 2. The client calls the Login method with invalid user name and valid password to login the application with Forms mode. |
| **Cleanup** | N/A |

MSAUTHWS\_S01\_TC02\_VerifyLoginUnderFormsAuthenticationWithInvalidUserName

|  |  |
| --- | --- |
| **S01\_LoginApplicationUnderFormsAuthentication** | |
| **Test case ID** | MSAUTHWS\_S01\_TC03\_VerifyLoginUnderFormsAuthenticationWithInvalidPassword |
| **Description** | This test case is used to verify the Login operation under Forms authentication mode is failed with invalid password. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode Forms. 2. The client calls the Login method with valid user name and invalid password to login the application with Forms mode. |
| **Cleanup** | N/A |

MSAUTHWS\_S01\_TC03\_VerifyLoginUnderFormsAuthenticationWithInvalidPassword

|  |  |
| --- | --- |
| **S02\_LoginApplicationUnderNoneAuthentication** | |
| **Test case ID** | MSAUTHWS\_S02\_TC01\_VerifyLoginUnderNoneAuthentication |
| **Description** | This test case is used to verify the Login operation under None authentication is failed. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode None. 2. The client calls the Login method with valid user name and password to login the application with None mode. |
| **Cleanup** | N/A |

MSAUTHWS\_S02\_TC01\_VerifyLoginUnderNoneAuthentication

|  |  |
| --- | --- |
| **S03\_LoginApplicationUnderWindowsAuthentication** | |
| **Test case ID** | MSAUTHWS\_S03\_TC01\_VerifyLoginUnderWindowsAuthentication |
| **Description** | This test case is used to verify the Login operation under Windows authentication is failed. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode Windows. 2. The client calls the Login method with valid user name and password to login the application with Windows mode. |
| **Cleanup** | N/A |

MSAUTHWS\_S03\_TC01\_VerifyLoginUnderWindowsAuthentication

|  |  |
| --- | --- |
| **S04\_LoginApplicationUnderPassportAuthentication** | |
| **Test case ID** | MSAUTHWS\_S04\_TC01\_VerifyLoginUnderPassportAuthentication |
| **Description** | This test case is used to verify the Mode and Login operations when the SUT's authentication mode is Passport. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. The client calls the Mode method to get the authentication mode Passport. 2. The client calls the Login method with valid user name and password to login the application with Passport mode. |
| **Cleanup** | N/A |

MSAUTHWS\_S04\_TC01\_VerifyLoginUnderPassportAuthentication