

MS-CPSWS Test Suite Specification

**Abstract:** This document provides information about how to configure the test suite and how MS-CPSWS test suite is designed to test MS-CPSWS 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](#_Toc387069094)

[1.1 Configuring the test suite client 3](#_Toc387069095)

[1.1.1 Configuring the test suite client manually 3](#_Toc387069096)

[1.1.2 Configuring the test suite client by scripts 3](#_Toc387069097)

[1.2 Configuring the system under test (SUT) 4](#_Toc387069098)

[1.2.1 Configuring the SUT manually 4](#_Toc387069099)

[1.2.2 Configuring the SUT by scripts 4](#_Toc387069100)

[1.3 Configuring the SHOULD/MAY requirements 4](#_Toc387069101)

[2 Test suite design 5](#_Toc387069102)

[2.1 Assumptions, scope and constraints 5](#_Toc387069103)

[2.2 Test suite architecture 5](#_Toc387069104)

[2.3 Technical dependencies and considerations 6](#_Toc387069105)

[2.4 Adapter design 7](#_Toc387069106)

[2.4.1 Adapter overview 7](#_Toc387069107)

[2.4.2 Technical feasibility of adapter approach 7](#_Toc387069108)

[2.4.3 Adapter abstract layer 8](#_Toc387069109)

[2.4.4 Adapter details 8](#_Toc387069110)

[2.5 Test scenarios 11](#_Toc387069111)

[2.5.1 S01\_RetrieveTypes 11](#_Toc387069112)

[2.5.2 S02\_RetrieveProviderHierarchyTree 11](#_Toc387069113)

[2.5.3 S03\_RetrieveProviderSchema 12](#_Toc387069114)

[2.5.4 S04\_ResolveToEntities 12](#_Toc387069115)

[2.5.5 S05\_SearchForEntities 13](#_Toc387069116)

[2.6 Test case design 14](#_Toc387069117)

[2.6.1 Traditional test case design 14](#_Toc387069118)

[2.6.2 Test case description 15](#_Toc387069119)

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

1. Open MS-CPSWS\TestSuite\MS-CPSWS\_TestSuite.deployment.ptfconfig file.
2. Update the following values 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="TargetHTTPServiceUrl" value="http://[SUTComputerName]/\_vti\_bin/SPClaimProviderWebService.svc"
* Property name="TargetHTTPSServiceUrl" value="https://[SUTComputerName]/\_vti\_bin/SPClaimProviderWebService.https.svc"
* Property name="HttpEndPointName" value="BasicHttpBinding\_IClaimProviderWebService"
* Property name="HttpsEndPointName" value="BasicHttpsBinding\_IClaimProviderWebService"

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

* Property name="OwnerLogin" value="[Domain]\[UserName]"
* Property name="ValidUser" value="contoso\cpswsuser1"
* Property name="MaxCount" value="10"
* Property name="NumberOfLevels" value="1"
* Property name="HierarchyProviderPrefix" value="\_HierarchyProvider\_"

### 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-CPSWS\_WindowsSharePointServices3\_SHOULDMAY.deployment.ptfconfig
* MS-CPSWS\_SharePointServer2007\_SHOULDMAY.deployment.ptfconfig
* MS-CPSWS\_SharePointFoundation2010\_SHOULDMAY.deployment.ptfconfig
* MS-CPSWS\_SharePointServer2010\_SHOULDMAY.deployment.ptfconfig
* MS-CPSWS\_SharePointFoundation2013\_SHOULDMAY.deployment.ptfconfig
* MS-CPSWS\_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-CPSWS\_SharePointFoundation2010\_SHOULDMAY.deployment.ptfconfig** and update the RXXXEnabled accordingly.

# Test suite design

## Assumptions, scope and constraints

Assumptions

This test suite assumes that the server is configured to have some claim providers by default.

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 13 operations: ClaimTypes, ClaimValueTypes, EntityTypes, GetHierarchy, GetHierarchyAll, HierarchyProviderSchema, ProviderSchemas, Resolve, ResolveClaim, ResolveMultiple, ResolveMultipleClaim, Search and SearchAll.
* 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 the operations over SOAP1.1.

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.



MS-CPSWS test suite architecture

The details of the MS-CPSWS test suite architecture

* SUT hosts the Sharepoint Claim Provider 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-CPSWS 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-CPSWS Open Specification.
* Test cases use the MS-CPSWS adapter to call and get the results of the MS-CPSWS operations.
* MS-CPSWS adapter is used in the test cases. The test cases call the methods in the interfaces to invoke the MS-CPSWS protocol adapter’s operations.
* The SUT control adapter is used in the test cases. The test cases call the methods in the interface to retrieve information from the SUT.

## Technical dependencies and considerations

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

Encryption consideration

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

## Adapter design

### Adapter overview

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

Protocol adapter

There are one protocol adapter: MS-CPSWS adapter, The protocol adapter is a managed adapters, which is derived from the ManagedAdapterBase class in PTF.

* MS-CPSWS adapter
* Chooses HTTP or HTTPS and SOAP 1.1 for transport;
* Constructs requests of 13 MS-CPSWS 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-CPSWS 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

* MS-CPSWS SUT control adapter
  + The SUT control adapter is a PowerShell script adapter; it is invoked by the test cases.
* The SUT control adapter has the following functionality:
* Get claim types, claim value types and entity types on the SUT.
* Make sure interactive mode can be used.

### Technical feasibility of adapter approach

Message generation

The MS-CPSWS adapter invoke the operations in proxy class of the protocol, which serializes the parameter values to the SOAP request messages, then the SOAP request messages are sent out by proxy class.

Message consumption

The messages received from the SUT will be parsed in the MS-CPSWS proxy class and be passed upon to the MS-CPSWS adapter. Then these messages are consumed in the MS-CPSWS adapter to validate the message format requirement, and consumed in the test cases to validate the logic-related requirements.

SUT control adapter

The SUT control adapter is designed to retrieve claim types, claim value types and entity types from the SUT.

### Adapter abstract layer

Protocol adapter

MS-CPSWS adapter interface

There are 13 methods declared in the MS-CPSWS adapter interface IMS\_CPSWSAdapter.

The methods are described in the following table:

|  |  |  |
| --- | --- | --- |
| No. | Methods | Description |
| 1 | ClaimTypes | Retrieve a list of all possible claim types from a list of claims providers. |
| 2 | ClaimValueTypes | Retrieve a list of claim value types from a list of claims providers. |
| 3 | EntityTypes | Retrieve a list of all possible picker entity types from a list of claims providers. |
| 4 | GetHierarchy | Retrieve a claims provider hierarchy tree from a claims provider. |
| 5 | GetHierarchyAll | Retrieve a list of claims provider hierarchy trees from a list of claims providers. |
| 6 | HierarchyProviderSchema | Retrieve schema for the current hierarchy provider. |
| 7 | ProviderSchemas | Retrieve a list of claims provider schemas from a list of claims providers. |
| 8 | Resolve | Resolve an input string to picker entities using a list of claims providers. |
| 9 | ResolveClaim | Resolve an input claim to picker entities using a list of claims providers. |
| 10 | ResolveMultiple | Resolve a list of strings to picker entities using a list of claims providers. |
| 11 | ResolveMultipleClaim | Resolve a list of claims to picker entities using a list of claims providers. |
| 12 | Search | Search for entities on a list of claims providers. |
| 13 | SearchAll | Search all entities on a list of claims providers. |

MS-CPSWS adapter interface methods

SUT control adapter

There are 3 methods defined in the SUT control adapter interface IMS\_CPSWSSUTControlAdapter.

The methods are described in the following table:

|  |  |  |
| --- | --- | --- |
| No. | Methods | Description |
| 1 | GetClaimTypesInSPProvider | Retrieve the claim types from the SUT. |
| 2 | GetClaimValueTypesInSPProvider | Retrieve the claim value types from the SUT. |
| 3 | GetEntityTypesInSPProvider | Retrieve the entity types from the SUT. |

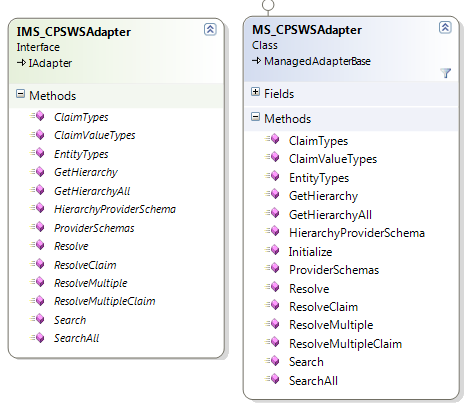
SUT control adapter interface method

### Adapter details

#### Protocol adapter

##### MS-CPSWS protocol adapter

The following figure shows the MS-CPSWS protocol adapter class diagram.



MS-CPSWS adapter class diagram

The following outlines details of the class diagram:

Adapter interface

* The IMS\_CPSWSAdapter is the interface of MS-CPSWSAdapter.
* The IMS\_CPSWSAdapter defines the methods invoked by test cases.

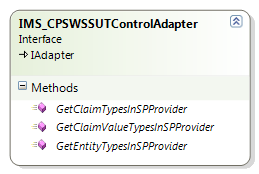
Adapter implementation

* MS-CPSWSAdapter is the protocol adapter class of the test suite. It is used to implement IMS\_CPSWSAdapter.
* The Initialize method is used to initialize the MS-CPSWS test suite.

#### SUT control adapter

##### MS-CPSWS SUT control adapter

The following figure shows the MS-CPSWS SUT control adapter class diagram.



SUT control adapter class diagram

The following outlines details of the class diagram:

The IMS\_CPSWSSUTControlAdapter 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

Five scenarios are designed to verify the server-side, testable requirements in MS-CPSWS test suite. The following table lists the scenarios used in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_RetrieveTypes | This scenario is designed to retrieve a list of all possible claim types, claim value types or entity types from a list of claim providers available to the protocol client. |
| S02\_RetrieveProviderHierarchyTree | This scenario is designed to retrieve provider hierarchy trees from a list of claim providers available to the protocol client. |
| S03\_RetrieveProviderSchema | This scenario is designed to retrieve provider schemas. |
| S04\_ResolveToEntities | This scenario is designed to resolve input strings/claims to picker entities using a list of claim providers. |
| S05\_SearchForEntities | This scenario is designed to search for entities on a list of claims providers. |
|

MS-CPSWS scenarios

### S01\_RetrieveTypes

Description

This scenario is designed to retrieve a list of all possible claim types, claim value types or entity types from a list of claim providers available to the protocol client.

Operations

* ClaimTypes
* ClaimValueTypes
* EntityTypes
* GetHierarchyAll

Prerequisites

N/A

Cleanup

N/A

### S02\_RetrieveProviderHierarchyTree

Description

This scenario is designed to retrieve provider hierarchy trees from a list of claim providers available to the protocol client.

Operations

* GetHierarchy
* GetHierarchyAll

Prerequisites

N/A

Cleanup

N/A

### S03\_RetrieveProviderSchema

Description

This scenario is designed to retrieve provider schemas.

Operations

* HierarchyProviderSchema
* ProviderSchemas

Prerequisites

N/A

Cleanup

N/A

### S04\_ResolveToEntities

Description

This scenario is designed to resolve input strings/claims to picker entities using a list of claim providers.

Operations

* Resolve
* ResolveClaim
* ResolveMultiple
* ResolveMultipleClaim

Prerequisites

N/A

Cleanup

N/A

### S05\_SearchForEntities

Description

This scenario is designed to search for entities on a list of claims providers.

Operations

* Search
* SearchAll
* HierarchyProviderSchema

Prerequisites

N/A

Cleanup

N/A

## 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.

31 traditional test cases are designed to cover the scenarios mentioned in [2.5 Test scenarios](#_Test_scenarios). Details of the traditional test cases are specified in [section 2.6.2 Test cases description](#_Test_cases_description). The scenarios distributions of the test cases are listed in the following table.

|  |  |
| --- | --- |
| Scenario ID | Test case name |
| S01\_RetrieveTypes | [MSCPSWS\_S01\_TC01\_ClaimTypes\_NullProviderNames](#S01_TC01) |
| [MSCPSWS\_S01\_TC02\_ClaimTypes\_AllValidProviderNames](#S01_TC02) |
| [MSCPSWS\_S01\_TC03\_ClaimTypes\_ValidProviderName](#S01_TC03) |
| [MSCPSWS\_S01\_TC04\_ClaimValueTypes\_NullProviderNames](#S01_TC04) |
| [MSCPSWS\_S01\_TC05\_ClaimValueTypes\_AllValidProviderNames](#S01_TC05) |
| [MSCPSWS\_S01\_TC06\_ClaimValueTypes\_ValidProviderNames](#S01_TC06) |
| [MSCPSWS\_S01\_TC07\_EntityTypes\_NullProviderNames](#S01_TC07) |
| [MSCPSWS\_S01\_TC08\_EntityTypes\_AllValidProviderNames](#S01_TC08) |
| [MSCPSWS\_S01\_TC09\_EntityTypes\_ValidProviderNames](#S01_TC09) |
| S02\_RetrieveProviderHierarchyTree | [MSCPSWS\_S02\_TC01\_GetHierarchy\_NullHierarchyNodeID](#S02_TC01) |
| [MSCPSWS\_S02\_TC02\_GetHierarchy\_ValidHierarchyNodeID](#S02_TC02) |
| [MSCPSWS\_S02\_TC03\_GetHierarchy\_ValidNumberOfLevels](#S02_TC03) |
| [MSCPSWS\_S02\_TC04\_GetHierarchy\_InvalidNumberOfLevels](#S02_TC04) |
| [MSCPSWS\_S02\_TC05\_GetHierarchyAll\_NullProviderNames](#S02_TC05) |
| [MSCPSWS\_S02\_TC06\_GetHierarchyAll\_AllOfProviderNames](#S02_TC06) |
| [MSCPSWS\_S02\_TC07\_GetHierarchyAll\_ValidInputParameters](#S02_TC07) |
| [MSCPSWS\_S02\_TC08\_GetHierarchyAll\_ValidNumberOfLevels](#S02_TC08) |
| [MSCPSWS\_S02\_TC09\_GetHierarchyAll\_InvalidNumberOfLevels](#S02_TC09) |
| S03\_RetrieveProviderSchema | [MSCPSWS\_S03\_TC01\_HierarchyProviderSchema](#S03_TC01) |
| [MSCPSWS\_S03\_TC02\_ProviderSchemas](#S03_TC02) |
| [MSCPSWS\_S03\_TC03\_ProviderSchemas\_NoInputProviderNames](#S03_TC03) |
| S04\_ResolveToEntities | [MSCPSWS\_S04\_TC01\_ResolveString](#S04_TC01) |
| [MSCPSWS\_S04\_TC02\_ResolveClaim\_Valid](#S04_TC02) |
| [MSCPSWS\_S04\_TC03\_ResolveMultipleStrings\_AllValid](#S04_TC03) |
| [MSCPSWS\_S04\_TC04\_ResolveMultipleStrings\_SomeValid](#S04_TC04) |
| [MSCPSWS\_S04\_TC05\_ResolveMultiple\_NullResolveInput](#S04_TC05) |
| [MSCPSWS\_S04\_TC06\_ResolveMultipleClaim\_SomeValid](#S04_TC06) |
| S05\_SearchForEntities | [MSCPSWS\_S05\_TC01\_Search](#S05_TC01) |
| [MSCPSWS\_S05\_TC02\_Search\_nullSearchPattern](#S05_TC02) |
| [MSCPSWS\_S05\_TC03\_SearchAll](#S05_TC03) |
| [MSCPSWS\_S05\_TC04\_SearchAll\_nullSearchPattern](#S05_TC04) |

Test case scenario distribution

Test cases are designed to verify the response messages and the core operations of this protocol.

### Test case description

There are 31 traditional test cases designed in this test suite.

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

The following tables describe the traditional test case.

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC01\_ClaimTypes\_NullProviderNames |
| **Description** | A test case used to test ClaimTypes method with providerNames parameter is set to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method ClaimTypes to get claim types with a null providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC01\_ClaimTypes\_NullProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC02\_ClaimTypes\_AllValidProviderNames |
| **Description** | A test case used to test ClaimTypes method with providerNames parameter is set to all of valid provider name. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of all provider. 2. Call protocol adapter method ClaimTypes to get claim types with all of valid providerNames in the request. 3. Call GetClaimTypesResultBySutAdapter method to get claim types with all of valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC02\_ClaimTypes\_AllValidProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC03\_ClaimTypes\_ValidProviderName |
| **Description** | A test case used to test ClaimTypes method with a valid providerNames parameter. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of each provider. 2. Call protocol adapter method ClaimTypes to get claim types with valid providerNames in the request. 3. Call GetClaimTypesResultBySutAdapter method to get claim types with valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC03\_ClaimTypes\_ValidProviderName

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC04\_ClaimValueTypes\_NullProviderNames |
| **Description** | A test case used to test ClaimValueTypes method with providerNames parameter is set to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method ClaimValueTypes to get claim value types with a null providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC04\_ClaimValueTypes\_NullProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC05\_ClaimValueTypes\_AllValidProviderNames |
| **Description** | A test case used to test ClaimValueTypes method with providerNames parameter is set to all of valid provider name. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of all provider. 2. Call protocol adapter method ClaimValueTypes to get claim value types with all of valid providerNames in the request. 3. Call GetClaimValueTypesResultBySutAdapter method to get claim value types with all of valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC05\_ClaimValueTypes\_AllValidProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC06\_ClaimValueTypes\_ValidProviderNames |
| **Description** | A test case used to test ClaimValueTypes method with a valid providerNames parameter. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of each provider. 2. Call protocol adapter method ClaimValueTypes to get claim value types with valid providerNames in the request. 3. Call GetClaimValueTypesResultBySutAdapter method to get claim value types with valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC06\_ClaimValueTypes\_ValidProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC07\_EntityTypes\_NullProviderNames |
| **Description** | A test case used to test EntityTypes method with providerNames parameter is set to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method EntityTypes to get entity types with a null providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC07\_EntityTypes\_NullProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC08\_EntityTypes\_AllValidProviderNames |
| **Description** | A test case used to test EntityTypes method with providerNames parameter is set to all of valid provider name. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of all provider. 2. Call protocol adapter method EntityTypes to get entity types with all of valid providerNames in the request. 3. Call GetEntityTypesResultBySutAdapter method to get entity types with all of valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC08\_EntityTypes\_AllValidProviderNames

|  |  |
| --- | --- |
| **S01\_RetrieveTypes** | |
| **Test case ID** | MSCPSWS\_S01\_TC09\_EntityTypes\_ValidProviderNames |
| **Description** | A test case used to test EntityTypes method with a valid providerNames parameter. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of each provider. 2. Call protocol adapter method EntityTypes to get entity types with valid providerNames in the request. 3. Call GetEntityTypesResultBySutAdapter method to get entity types with valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S01\_TC09\_EntityTypes\_ValidProviderNames

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC01\_GetHierarchy\_NullHierarchyNodeID |
| **Description** | A test case used to test GetHierarchy method with hierarchyNodeID parameter is set to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set the valid number of levels of claims provider hierarchy trees by configuration. 2. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, and get the provider name of the first provider with a child. 3. Call protocol adapter method GetHierarchy to get a claims provider hierarchy tree with a null hierarchyNodeID in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC01\_GetHierarchy\_NullHierarchyNodeID

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC02\_GetHierarchy\_ValidHierarchyNodeID |
| **Description** | A test case used to test GetHierarchy method with a valid hierarchyNodeID parameter. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get the provider name of the first provider with a child which hierarchyNodeID is not null. 2. Set the valid number of levels of claims provider hierarchy trees by configuration. 3. Call protocol adapter method GetHierarchy to get a claims provider hierarchy tree with a valid child hierarchyNodeID in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC02\_GetHierarchy\_ValidHierarchyNodeID

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC03\_GetHierarchy\_ValidNumberOfLevels |
| **Description** | A test case used to test GetHierarchy method with a valid numberOfLevels parameter. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set the valid number of levels of claims provider hierarchy trees by configuration. 2. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of each provider. 3. Call protocol adapter method GetHierarchy to get a claims provider hierarchy tree with a valid numberOfLevels parameter in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC03\_GetHierarchy\_ValidNumberOfLevels

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC04\_GetHierarchy\_InvalidNumberOfLevels |
| **Description** | A test case used to test GetHierarchy method with numberOfLevels parameter is less than 1. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get the first provider with provider name is not null and set its provider name as GetHierarchy input parameter. 2. Call protocol adapter method GetHierarchy with numberOfLevels parameter sets to invalid. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC04\_GetHierarchy\_InvalidNumberOfLevels

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC05\_GetHierarchyAll\_NullProviderNames |
| **Description** | A test case used to test GetHierarchyAll method with providerNames parameter is set to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set the valid number of levels of claims provider hierarchy trees by configuration. 2. Call protocol adapter method GetHierarchyAll to get a list of claims provider hierarchy trees with a null providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC05\_GetHierarchyAll\_NullProviderNames

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC06\_GetHierarchyAll\_AllOfProviderNames |
| **Description** | A test case used to test GetHierarchyAll method with providerNames parameter is set to all of valid provider name. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of all provider. 2. Set the valid number of levels of claims provider hierarchy trees by configuration. 3. Call protocol adapter method GetHierarchyAll to get a list of claims provider hierarchy trees with all of valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC06\_GetHierarchyAll\_AllOfProviderNames

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC07\_GetHierarchyAll\_ValidInputParameters |
| **Description** | A test case used to test GetHierarchyAll method with valid input parameters. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of each provider. 2. Set the valid number of levels of claims provider hierarchy trees by configuration. 3. Call protocol adapter method GetHierarchyAll to get a list of claims provider hierarchy trees with a valid providerNames in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC07\_GetHierarchyAll\_ValidInputParameters

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC08\_GetHierarchyAll\_ValidNumberOfLevels |
| **Description** | A test case used to test GetHierarchyAll method with valid value of numberOfLevels. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Set the valid number of levels of claims provider hierarchy trees by configuration. 2. Call protocol adapter method GetHierarchy to get a claims provider hierarchy tree with a valid numberOfLevels parameter and provider names is set to null in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC08\_GetHierarchyAll\_ValidNumberOfLevels

|  |  |
| --- | --- |
| **S02\_RetrieveProviderHierarchyTree** | |
| **Test case ID** | MSCPSWS\_S02\_TC09\_GetHierarchyAll\_InvalidNumberOfLevels |
| **Description** | A test case used to test GetHierarchyAll method with numberOfLevels parameter is less than 1. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers on the site, get provider name of all provider. 2. Call protocol adapter method GetHierarchyAll with numberOfLevels parameter sets to invalid. |
| **Cleanup** | N/A |

MSCPSWS\_S02\_TC09\_GetHierarchyAll\_InvalidNumberOfLevels

|  |  |
| --- | --- |
| **S03\_RetrieveProviderSchema** | |
| **Test case ID** | MSCPSWS\_S03\_TC01\_HierarchyProviderSchema |
| **Description** | A method used to verify retrieving the schema of the current hierarchy provider successfully. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method HierarchyProviderSchema to retrieve the schema of the current hierarchy provider. |
| **Cleanup** | N/A |

MSCPSWS\_S03\_TC01\_HierarchyProviderSchema

|  |  |
| --- | --- |
| **S03\_RetrieveProviderSchema** | |
| **Test case ID** | MSCPSWS\_S03\_TC02\_ProviderSchemas |
| **Description** | A method used to verify retrieving the schemas of the specific claims providers successfully. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method ProviderSchemas to get schemas of the claims providers specified in the list of provider names in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S03\_TC02\_ProviderSchemas

|  |  |
| --- | --- |
| **S03\_RetrieveProviderSchema** | |
| **Test case ID** | MSCPSWS\_S03\_TC03\_ProviderSchemas\_NoInputProviderNames |
| **Description** | A method used to verify retrieving schemas of all the available claims providers by not specify provider names in the request. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method ProviderSchemas to get schemas of all the available claims providers by not specify provider names in the request. |
| **Cleanup** | N/A |

MSCPSWS\_S03\_TC03\_ProviderSchemas\_NoInputProviderNames

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC01\_ResolveString |
| **Description** | A test case used to test resolve method with valid input string. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method Resolve to resolve an input string to picker entities using a list of claims providers. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC01\_ResolveString

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC02\_ResolveClaim\_Valid |
| **Description** | This test case is used to test typical resolve claim scenario. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method ResolveClaim to resolve an SPClaim to picker entities using a list of claims providers. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC02\_ResolveClaim\_Valid

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC03\_ResolveMultipleStrings\_AllValid |
| **Description** | This test case is used to resolve 2 valid users to picker entities. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method ResolveMultiple to resolve 2 valid users to picker entities using a list of claims providers. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC03\_ResolveMultipleStrings\_AllValid

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC04\_ResolveMultipleStrings\_SomeValid |
| **Description** | This test case is used to resolve 2 users to picker entities, one is valid and another is invalid. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method ResolveMultiple to resolve 2 users to picker entities, one valid and another invalid. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC04\_ResolveMultipleStrings\_SomeValid

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC05\_ResolveMultiple\_NullResolveInput |
| **Description** | This test case is used test resolve multiple method with resolveInput parameter sets to null. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method ResolveMultiple with resolveInput parameter sets to null. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC05\_ResolveMultiple\_NullResolveInput

|  |  |
| --- | --- |
| **S04\_ResolveToEntities** | |
| **Test case ID** | MSCPSWS\_S04\_TC06\_ResolveMultipleClaim\_SomeValid |
| **Description** | This test case is used test resolve 2 claims to picker entities, one valid and another invalid. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all valid claim providers. 2. Call protocol adapter method ResolveMultipleClaim to resolve 2 claims to picker entities using a list of claims providers. |
| **Cleanup** | N/A |

MSCPSWS\_S04\_TC06\_ResolveMultipleClaim\_SomeValid

|  |  |
| --- | --- |
| **S05\_SearchForEntities** | |
| **Test case ID** | MSCPSWS\_S05\_TC01\_Search |
| **Description** | A method used to verify the search operation can successfully find the first claims provider tree that contains a child. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get the first provider with a child and set its provider name and Hierarchy Node ID as search inputs. Use the child’s display name (Nm) as the searchPattern. 2. Set the max count of the matching entities by configuration. 3. Call protocol adapter method Search with SecurityGroup as principalType, and the search inputs. 4. Call protocol adapter method HierarchyProviderSchema to identify the hierarchy provider name, if any. |
| **Cleanup** | N/A |

MSCPSWS\_S05\_TC01\_Search

|  |  |
| --- | --- |
| **S05\_SearchForEntities** | |
| **Test case ID** | MSCPSWS\_S05\_TC02\_Search\_nullSearchPattern |
| **Description** | A method used to verify when searchPattern is null as search input, the server will return an ArgumentNullException (searchPattern) message. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method Search with searchPattern as null. |
| **Cleanup** | N/A |

MSCPSWS\_S05\_TC02\_Search\_nullSearchPattern

|  |  |
| --- | --- |
| **S05\_SearchForEntities** | |
| **Test case ID** | MSCPSWS\_S05\_TC03\_SearchAll |
| **Description** | A method used to verify the SearchAll operation can successfully find the first claims provider tree that contains a child. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method GetHierarchyAll to get all the available providers on the site, and get the first provider with a child and use the child’s display name (Nm) as the searchPattern. 2. Set the max count of the matching entities by configuration. 3. Call protocol adapter method SearchAll with SecurityGroup as principalType, and use all the available providers in the providerNames input parameter. 4. Call protocol adapter method HierarchyProviderSchema to identify the hierarchy provider name, if any. |
| **Cleanup** | N/A |

MSCPSWS\_S05\_TC03\_SearchAll

|  |  |
| --- | --- |
| **S05\_SearchForEntities** | |
| **Test case ID** | MSCPSWS\_S05\_TC04\_SearchAll\_nullSearchPattern |
| **Description** | A method used to verify when searchPattern is null as the SearchAll input, the server will return an ArgumentNullException(searchPattern) message. |
| **Prerequisites** | N/A |
| **Test execution steps** | 1. Call protocol adapter method SearchAll with searchPattern as null. |
| **Cleanup** | N/A |

MSCPSWS\_S05\_TC04\_SearchAll\_nullSearchPattern