

**Exchange MAPI Test Suites Specification**

**Contents**

[1 Introduction 3](#_Toc400796655)

[2 Requirement specification 4](#_Toc400796656)

[3 Design considerations 5](#_Toc400796657)

[3.1 Assumptions 5](#_Toc400796658)

[3.2 Dependencies 5](#_Toc400796659)

[4 Package design 6](#_Toc400796660)

[4.1 Architecture 6](#_Toc400796661)

[4.2 Common library 7](#_Toc400796662)

[4.2.1 Transport classes 7](#_Toc400796663)

[4.2.2 Helper methods 7](#_Toc400796664)

[4.2.3 Message structures 7](#_Toc400796665)

[4.3 Adapter 7](#_Toc400796666)

[4.3.1 Protocol Adapter 7](#_Toc400796667)

[4.3.2 SUT Control Adapter 8](#_Toc400796668)

[4.4 Model 8](#_Toc400796669)

[4.4.1 Actions 8](#_Toc400796670)

[4.4.2 States 8](#_Toc400796671)

[4.4.3 Machines 8](#_Toc400796672)

[4.4.4 Model design patterns 9](#_Toc400796673)

[4.5 Test suite 9](#_Toc400796674)

[4.5.1 MS-OXCFOLD 9](#_Toc400796675)

[4.5.2 MS-OXCFXICS 9](#_Toc400796676)

[4.5.3 MS-OXCMAPIHTTP 10](#_Toc400796677)

[4.5.4 MS-OXCMSG 10](#_Toc400796678)

[4.5.5 MS-OXCNOTIF 11](#_Toc400796679)

[4.5.6 MS-OXCPERM 11](#_Toc400796680)

[4.5.7 MS-OXCPRPT 11](#_Toc400796681)

[4.5.8 MS-OXCROPS 12](#_Toc400796682)

[4.5.9 MS-OXCRPC 12](#_Toc400796683)

[4.5.10 MS-OXCSTOR 12](#_Toc400796684)

[4.5.11 MS-OXCTABL 13](#_Toc400796685)

[4.5.12 MS-OXNSPI 13](#_Toc400796686)

[4.5.13 MS-OXORULE 13](#_Toc400796687)

# Introduction

The Exchange MAPI Protocol Test Suites are implemented as synthetic clients running against a server-side implementation of a given Exchange protocol. They are designed in a client-to-server relationship and were originally developed for the in-house testing of the Microsoft Open Specifications. Test suites have been used extensively in Plugfests and Interoperability Labs to test partner implementations.

This document describes how the Exchange MAPI Protocol Test Suites are designed to verify that the server behaves in the way that is compliant with normative protocol requirements as described in the technical specification.

The Microsoft Open Specifications were written using the normative language defined in [RFC2119](http://go.microsoft.com/fwlink/?LinkId=117453). The statements of them are extracted as protocol requirements which are listed in the requirement specification described in section 2. The test suites are developed to test the normative protocol requirements. In a single test suite, similar or related requirements are grouped into one test case, and the test cases about same command or operation are grouped into one scenario.

The technical specifications listed in the following table are included in the Exchange MAPI Protocol Test Suites package. The version of these technical specifications is v20140130.

Exchange MAPI Protocol technical specifications

|  |  |  |
| --- | --- | --- |
| Technical specification | | Protocol name |
| MS-OXCFOLD | [Folder Object Protocol](http://go.microsoft.com/fwlink/?LinkID=267166) | |
| MS-OXCFXICS | [Bulk Data Transfer Protocol](http://go.microsoft.com/fwlink/?LinkID=267170) | |
| MS-OXCMAPIHTTP | [Messaging Application Programming Interface (MAPI) Extensions for HTTP](http://go.microsoft.com/fwlink/?LinkID=396883) | |
| MS-OXCMSG | [Message and Attachment Object Protocol](http://go.microsoft.com/fwlink/?LinkID=267172) | |
| MS-OXCNOTIF | [Core Notifications Protocol](http://go.microsoft.com/fwlink/?LinkID=330226) | |
| MS-OXCPERM | [Exchange Access and Operation Permissions Protocol](http://go.microsoft.com/fwlink/?LinkID=267175) | |
| MS-OXCPRPT | [Property and Stream Object Protocol](http://go.microsoft.com/fwlink/?LinkID=267174) | |
| MS-OXCROPS | [Remote Operations (ROP) List and Encoding Protocol](http://go.microsoft.com/fwlink/?LinkID=267173) | |
| MS-OXCRPC | [Wire Format Protocol](http://go.microsoft.com/fwlink/?LinkID=267171) | |
| MS-OXCSTOR | [Store Object Protocol](http://go.microsoft.com/fwlink/?LinkID=267169) | |
| MS-OXCTABL | [Table Object Protocol](http://go.microsoft.com/fwlink/?LinkID=267168) | |
| MS-OXNSPI | [Exchange Server Name Service Provider Interface (NSPI) Protocol](http://go.microsoft.com/fwlink/?LinkId=330225) | |
| MS-OXORULE | [E-Mail Rules Protocol](http://go.microsoft.com/fwlink/?LinkID=267176) | |

# Requirement specification

A requirement specification contains a list of requirements that are extracted from statements in the technical specification. Each technical specification has one corresponding requirement specification named as MS-XXXX\_RequirementSpecification.xlsx, which can be found in the Docs\MS-XXXX folder in the Exchange MAPI Protocol Test Suites package together with the technical specification.

The requirements are categorized as normative or informative. If the statement of the requirement is required for interoperability, the requirement is normative. If the statement of the requirement is clarifying information or high-level introduction, and removal of it does not affect interoperability, the requirement is informative.

Each requirement applies to a specific scope: server, client, or both. If the requirement describes a behavior performed by the responder, the scope of the requirement is server. If the requirement describes a behavior performed by the initiator, the scope of the requirement is client. If the requirement describes a behavior performed by both initiator and responder, the scope of the requirement is both.

The test suites cover normative requirements which describes a behavior performed by the responder. For a detailed requirements list and classification, see the MS-XXXX\_RequirementSpecification.xlsx.

# Design considerations

## Assumptions

* The Exchange MAPI Protocol Test Suites are not designed to run multi-protocol user scenarios, but rather provide a way to exercise certain operations documented in a technical specification.
* The test suites are functional tests that verify the compatibility of the system under test (SUT) with a protocol implementation.
* The test suites do not cover every protocol requirement and in no way certify an implementation, even if all tests pass.
* The test suites verify the server-side testable requirements; they do not verify the requirements related to client behaviors and server internal behaviors.

## Dependencies

* All Exchange MAPI Protocol Test Suites depend on the Protocol Test Framework (PTF) to derive managed adapters.
* MS-OXCFXICS, MS-OXCPRPT and MS-OXCTABL protocol test suites depend on the Spec Explorer to generate the test suites code based on the models.

# Package design

The Exchange MAPI Protocol Test Suites are implemented as synthetic clients running against a server-side implementation of a given Exchange protocol. The test suites verify the server-side and testable requirements.

## Architecture

The following figure illustrates the Exchange MAPI Protocol Test Suites architecture.



**Figure 1: Architecture**

The following outlines the details of the test suites architecture:

**SUT**

The system under test (SUT) hosts the server-side implementation of the protocol, which test suites run against.

* + From a third-party’s point of view, the SUT is a server implementation.
  + The following products have been tested with the test suites on the Windows platform.
* Microsoft Exchange Server 2007 Service Pack 3 (SP3)
* Microsoft Exchange Server 2010 Service Pack 3 (SP3)
* Microsoft Exchange Server 2013 Service Pack 1 (SP1)
* Microsoft Exchange Server 2016

The presence of a second SUT will enable certain multi-server scenarios and other test cases that pertain to a second SUT. If the second SUT is not present, then some steps of the test cases will not be run.

**Test Suite Client**

The test suites act as synthetic clients to communicate with the SUT and validate the requirements gathered from technical specifications. The Exchange MAPI Protocol Test Suites include one common library, 13 adapters, 13 test suites, and three models.

* The test suites communicate with SUT via a protocol adapter and SUT control adapter to verify if the SUT behaves in the way that is compliant with normative protocol requirements.
* The MS-OXNSPI, MS-OXCRPC and MS-OXCMAPIHTTP protocol adapters communicate directly with the SUT. All other protocol adapters communicate with SUT through the common library.

## Common library

The common library provides implementation of the transport layer, common messages, structures, and helper methods.

### Transport classes

There are three transport classes in the common library: MapiHttpAdapter, RpcAdapter and OxcropsClient.

The MapiHttpAdapter transport class implements HTTP communications between test suites and SUTs.

The RpcAdapter transport class implements RPC communications by calling native methods generated from IDL.

The MapiHttpAdapter transport class or RpcAdapter transport class is chosen according to the setting of the “TransportSeq” property in the ExchangeCommonConfiguration.deployment.ptfconfig file. If the “TransportSeq” property is set as “mapi\_http”, the MapiHttpAdapter transport class is chosen to send and receive message with MS-OXCMAPIHTTP transport; If the “TransportSeq” property is set as either “ncacn\_ip\_tcp” or “ncacn\_http”, the RpcAdapter transport class is chosen to send and receive messages with MS-OXCRPC transport.

The OxcropsClient works as an intermediary between the protocol adapter and MapiHttpAdapter/RpcAdapter class. The protocol adapter calls OxcropsClient to connect, disconnect, and run ROP commands. The Oxcropsclient passes the protocol adapter requests to the MapiHttpAdapter/RpcAdapter, and the MapiHttpAdapter/RpcAdapter sends the protocol adapter request to the SUT. The MapiHttpAdapter/RpcAdapter receives the response from the SUT and passes the response to OxcropsClient, and the OxcropsClient sends the response back to the protocol adapter.

### Helper methods

The common library defines a series of helper methods. The helper methods can be classified into following categories.

* Access the properties in the configuration file.
* Generate resource name.
* Other methods which are used by multiple test suites.

### Message structures

Except the MS-OXMAPIHTTP, MS-OXNSPI and MS-OXCRPC, all other protocols are ROP protocols. Therefore the ROP message structures are used by multiple test suites. These ROP message structures are defined in the common library.

## Adapter

Adapters are interfaces between the test suites and the SUT. There are two types of adapter: protocol adapter and SUT control adapter. In most cases, modifications to the protocol adapter will not be required for non-Microsoft SUT implementations. However, the SUT control adapter should be appropriately configured to connect to a non-Microsoft SUT implementation. All test suites in the package contain a protocol adapter, six of them contain a SUT control adapter.

### Protocol Adapter

The protocol adapter is a managed adapter, which is derived from the ManagedAdapterBase class in the PTF. It provides an interface that is used by the test cases to construct protocol request messages that will be sent to the SUT. The protocol adapter also acts as an intermediary between the test cases and the transport classes, receiving messages, sending messages, parsing responses from the transport classes, and validating the SUT response according to the normative requirement in the technical specification.

Except MS-OXMAPIHTTP, MS-OXNSPI and MS-OXCRPC, all other protocol adapters use MapiHttpAdapter and RpcAdapter transport classes defined in the common library to send and receive messages. MS-OXMAPIHTTP, MS-OXNSPI, and MS-OXCRPC protocol adapters communicate directly with the SUT.

### SUT Control Adapter

The SUT control adapter manages all the control functions of the test suites that are not associated with the protocol. For example, the setup and tear down are managed through the SUT control adapter (that is, enabling or disabling the asynchronous RPC notification on the SUT). The SUT control adapter is designed to work with the Microsoft implementation of the SUT. However, it is configurable to allow the test suites to run against non-Microsoft implementations of the SUT.

There are six protocols that have a SUT control adapter in the Exchange MAPI Protocol test suites package: MS-OXCMAPIHTTP, MS-OXCPERM, MS-OXCSTOR, MS-OXCRPC, MS-OXNSPI, and MS-OXORULE.

## Model

There are three model-based test suites in the Exchange MAPI Protocol Test Suites package: MS-OXCFXICS, MS-OXCRPRPT, and MS-OXCTABL. In the model-based test suite, the protocol behaviors are modeled by defining actions, states, and machines. The machines are used to generate test suite source code through Spec Explorer.

### Actions

The actions are abstracted from the protocol operations specified in the technical specification. All the protocol operations have corresponding actions. Besides the protocol operations, the models in the Exchange MAPI Protocol Test Suites package also define the actions which check whether the requirement is enabled for a given Exchange Server version, and the actions which check whether the MS-OXCMAPIHTTP transport is supported for a given Exchange Server version.

The actions are defined as a series of rule methods with action attribute in the model class. In the adapter, there is a corresponding adapter method for each action. The adapter method will transform the abstractive parameters of the actions to actual parameters and send the real request.

### States

The models in the Exchange MAPI Protocol Test Suites package define a serial of states to simulate the state change of the SUT. The states are defined as static variables in the model class. Each action checks the current state to decide its appropriate operation, and it also will update the state if the SUT state is changed.

### Machines

The models in the Exchange MAPI Protocol Test Suites package define a serial of machines to simulate the protocol operation sequences defined in the technical specification, and the SUT state transition after the SUT receives the request from the test suite client.

There are three kinds of machines are defined in the Cord file of the model project: scenario machines, sliced machines, and test suite machines. Scenario machines are designed to model the scenario defined in section 4.5. Sliced machines are designed to slice the scenario machines to be testable machines by limiting the parameters for each operation in the scenario machines. Test suite machines are defined for test cases generation. They construct the test cases by applying “ShortTests” strategy to sliced machines.

### Model design patterns

Five patterns are used in the models of the Exchange MAPI Protocol Test Suites package including SynchronousProtocols, AbstractIdentifiers, Server Initialization Pattern, Check Return Value, and Helper File Pattern.

* SynchronousProtocols: The protocol operations are modeled as synchronous behavior where the test suite client sends a request and waits for a response. A request is always immediately followed by its response, without interleaving packets from other interchanges.
* AbstractIdentifiers: The models explore a set of simple types (Boolean, enumeration) as placeholders for the concrete values that will appear during protocol testing. For example, the actual values of CodePage GUID cannot be identified at modeling time, so these variables are replaced by abstract identifiers and the real data is stored in the protocol adapter.
* Server Initialization Pattern: The models make a connection to the server and log on to a private mailbox/public folder.
* Check Return Value Pattern: The models verify protocol model actions’ return values related to some requirements that are verified by the model test case. The capture codes are written in corresponding actions in the protocol model.
* Helper File Pattern: The models add a new file, ModelHelper.cs in the Model project, and design the capture requirement method in the ModelHelper.cs file.

## Test suite

The test suites verify the server-side and testable requirements listed in the requirement specification. The test suites call the protocol adapter to send and receive message between the protocol adapter and the SUT, and call the SUT control adapter to change the SUT state. The test suites consists of a series test cases which are categorized to several scenarios. Some test cases rely on a second SUT. If the second SUT is not present, then some steps of these test cases will not be run.

### MS-OXCFOLD

Five scenarios are designed to verify the server-side, testable requirements in MS-OXCFOLD test suite. The S04\_OperateOnPublicFolder scenario relies on the second SUT. It will not be run if the second SUT is not available. The following table lists the scenarios designed in the test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_FolderRopOperations | Verifies the ROP operations related to a general folder object. |
| S02\_MessageRopOperations | Verifies the ROP operations related to messages or subfolders in a folder object. |
| S03\_FolderInformation | Verifies the properties contained in a folder object or ROP operations related to a search folder. |
| S04\_OperateOnPublicFolder | Verifies the ROP operations on public folders. |
| S05\_InsufficientRightsOnFolder | Verifies the ROP operations that the client has insufficient rights to operate on the specified private mailbox folder. |

### MS-OXCFXICS

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SyncFastTransferFolder | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to synchronizing content of folders between the client and the server. |
| S02\_SyncFastTransferMessage | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to synchronizing content of messages between the client and the server. |
| S03\_SyncFastTransferAttachment | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to synchronizing content of attachments between the client and the server. |
| S04\_SyncFastTransferBetweenServers | Tests the ROPs and server behavior related to synchronizing data between different servers. |
| S05\_SyncICSHierarchy | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to hierarchy synchronizations. |
| S06\_SyncICSContents | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to contents synchronization. |
| S07\_SyncICSState | Tests the ROPs, properties, structures, FastTransfer stream format and the server behavior related to ICS state synchronizations. |
| S08\_SyncICSFolderConfilict | Tests the ROPs, properties and the server behavior related to ICS folder conflict. |
| S09\_SyncICSMessageConflict | Tests the ROPs, properties and the server behavior related to ICS message conflict. |

### MS-OXCMAPIHTTP

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_RequestTypesForMailboxServerEndpoint | Verifies the HTTP header, common response format, and the request types for mailbox server endpoint. |
| S02\_RequestTypesForAddressBookServerEndpoint | Verifies the request types for address book server endpoint. |

### MS-OXCMSG

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateAndSaveMessage | Validates the behaviors when the client calls the Message methods to create and save Message objects on the server. Finally, the client releases the Message object by calling the RopRelease operation. |
| S02\_SetMessageStatus | Validates the behaviors when the client calls the Message methods to set and get the message status. Finally, the client releases the Message object by calling the RopRelease operation. |
| S03\_SetMessageFlags | Validates the behaviors when the client calls the Message methods to set and get read flags of a message. Finally, the client releases the Message object by calling the RopRelease operation. |
| S04\_MessageObjectProperties | Validates the Message object properties on the server. Finally, the client releases the Message object by calling the RopRelease operation. |
| S05\_OpenMessage | Validates the behaviors when the client calls the Message methods to create, save, and open Message objects on the server. Finally, the client releases the Message object by calling the RopRelease operation. |
| S06\_ReloadCachedInformation | Validates the behaviors when the client calls the Message methods to open and reload Message objects on the server. Finally, the client releases the Message object by calling the RopRelease operation. |
| S07\_RopRecipient | Validates the behaviors when the client calls the Recipient methods to read, modify, and remove all recipients on a Message object. Finally, the client releases the Message object by calling the RopRelease operation. |
| S08\_RopAttachment | Validates the behaviors when the client calls the Attachment methods to open, get attachment tables, and delete attachment on a Message object. The client releases the Attachment object and Message object by calling RopRelease operation. |
| S09\_RopEmbeddedMessage | Validates the behaviors when the client calls the Embedded Message methods to create, save, and open an embedded Message object in an Attachment object on the server. |

### MS-OXCNOTIF

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_ReceivePendingNotifications | Verifies the four ways that the server uses to notify the client of pending notifications. |
| S02\_SubscribeAndReceiveNotifications | Verifies the notification details returned from the server after the client subscribes to the events on the server. |

### MS-OXCPERM

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_RetrieveFolderPermissions | Verifies the response for retrieving the folder permissions request. |
| S02\_ModifyFolderPermissions | Verifies the response for modifying the folder permissions request. |
| S03\_NegativeOrErrorValidation | Verifies the responses for the wrong message sequences and the negative behaviors. |

### MS-OXCPRPT

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_QueryDataFromObject | Validates the operations used for querying data from an object. |
| S02\_SetDataForObject | Validates the operations used for [setting data](#S2) for an object. |
| S03\_QueryDataFromStreamObject | Validates the operations used for querying data of stream object. |
| S04\_SetDataForStreamObject | Validates the operations for setting data for stream object. |
| S05\_AsynchronousScenario | Validates the operations used for asynchronous transition. |
| S06\_TestCommonObjectProperties | Validates the common object properties. |

### MS-OXCROPS

12 scenarios are designed for this test suite to verify the server-side, testable requirements in MS-OXCROPS test suite. The scenario S01\_LogonROPs relies on the second SUT. Some steps of the test cases will not be run if the second SUT is not available. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_LogonROPs | Verifies the response buffer formats of Logon ROPs. |
| S02\_FolderROPs | Verifies the response buffer formats of Folder ROPs. |
| S03\_TableROPs | Verifies the response buffer formats of Table ROPs. |
| S04\_MessageROPs | Verifies the response buffer formats of Message ROPs. |
| S05\_TransportROPs | Verifies the response buffer formats of Transport ROPs. |
| S06\_PropertyROPs | Verifies the response buffer formats of Property ROPs. |
| S07\_StreamROPs | Verifies the response buffer formats of Stream ROPs. |
| S08\_PermissionROPs | Verifies the response buffer formats of Permission ROPs. |
| S09\_RuleROPs | Verifies the response buffer formats of Rule ROPs. |
| S10\_FastTransferROPs | Verifies the response buffer formats of Fast Transfer ROPs. |
| S11\_IncrementalChangeSynchronizationROPs | Verifies the response buffer formats of Incremental Change Synchronization ROPs. |
| S12\_NotificationROPs | Verifies the response buffer formats of Notification ROPs and RopBufferTooSmall ROP. |

### MS-OXCRPC

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SynchronousCall | Verifies the requirements related to the EMSMDB interface. |
| S02\_AsynchronousCall | Verifies the requirements related to the AsyncEMSMDB interface. |

### MS-OXCSTOR

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_PrivateMailboxLogon | Validates the operations performed against a private mailbox logon. |
| S02\_PublicFoldersLogon | Validates the operations performed against a public folder logon. |
| S03\_SyncUpReadAndUnreadInformation | Validates the operations for clients synchronize Per-User Read/Unread data. |

### MS-OXCTABL

Seven scenarios are designed to verify the server-side and testable requirements in MS-OXCTABL test suite. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_ColumnRops | Tests the ROPs for processing table columns in MS-OXCTABL. |
| S02\_RowRops | Tests the ROPs for processing table rows in MS-OXCTABL. |
| S03\_BookmarkRops | Tests the ROPs on a table bookmark in MS-OXCTABL. |
| S04\_ExpandRowRops | Tests the ROPs for expanding or collapsing a categorized row in MS-OXCTABL. |
| S05\_ResetSortTableRops | Tests the ROPs for resetting or sorting a table in MS-OXCTABL. |
| S06\_RestrictRop | Tests the ROP for restricting a table in MS-OXCTABL. |
| S07\_AsyncRops | Tests the asynchronous ROPs in MS-OXCTABL. |

### MS-OXNSPI

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_ObtainGeneralInfo | Tests the server behavior for the NSPI calls related to obtaining general information of Address Book object. |
| S02\_ObtainDetailsInfo | Tests the server behavior for the NSPI calls related to obtaining the detailed information of Address Book object. |
| S03\_ANRRelatedBehavior | Tests the server behavior for the NSPI calls related to Ambiguous Name Resolution process. |
| S04\_ModifyProperty | Tests the server behavior for the NSPI calls related to modify property of Address Book object. |
| S05\_NegativeBehavior | Tests the negative server behavior for each NSPI call. |

### MS-OXORULE

Five scenarios are designed for the MS-OXORULE test suite to verify the structure and the server behavior. The following table lists the scenarios designed in this test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_AddModifyDeleteRetrieveRules | Validates server behaviors of the following:   * The operations of RopModifyRules and RopGetRulesTable when adding, modifying, deleting and retrieving standard rules. * The functions of ROPs referenced from MS-OXCMSG and MS-OXCTABL for adding, modifying, deleting, and retrieving extended rules. |
| S02\_ProcessServerSideRulesOtherthanOOF | Validates server behaviors of processing server-side rules other than the Out-of-Office rule because the action of OP\_OOF\_REPLY is complicated enough to be a separate scenario. |
| S03\_ProcessOutOfOfficeRule | Validates server behaviors of processing the Out-of-Office rule. |
| S04\_ProcessRulesOnPublicFolder | Validates server behaviors of processing server-side rules on the public folder. |
| S05\_GenerateDAMAndDEM | Validates server behaviors about DAM and DEM message. |