

**Exchange EWS Test Suites Specification**

**Contents**

[1 Introduction 3](#_Toc421794289)

[2 Requirement specification 4](#_Toc421794290)

[3 Design considerations 5](#_Toc421794291)

[3.1 Assumptions 5](#_Toc421794292)

[3.2 Dependencies 5](#_Toc421794293)

[4 Package design 6](#_Toc421794294)

[4.1 Architecture 6](#_Toc421794295)

[4.2 Common library 7](#_Toc421794296)

[4.2.1 Transport classes 7](#_Toc421794297)

[4.2.2 Helper methods 7](#_Toc421794298)

[4.2.3 Message structures 7](#_Toc421794299)

[4.3 Adapter 7](#_Toc421794300)

[4.3.1 Protocol Adapter 7](#_Toc421794301)

[4.3.2 SUT Control Adapter 7](#_Toc421794302)

[4.4 Test suite 8](#_Toc421794303)

[4.4.1 MS-OXWSATT 8](#_Toc421794304)

[4.4.2 MS-OXWSBTRF 8](#_Toc421794305)

[4.4.3 MS-OXWSCONT 8](#_Toc421794306)

[4.4.4 MS-OXWSCORE 8](#_Toc421794307)

[4.4.5 MS-OXWSFOLD 9](#_Toc421794308)

[4.4.6 MS-OXWSMSG 9](#_Toc421794309)

[4.4.7 MS-OXWSMTGS 10](#_Toc421794310)

[4.4.8 MS-OXWSSYNC 10](#_Toc421794311)

[4.4.9 MS-OXWSTASK 11](#_Toc421794312)

# Introduction

The Exchange EWS 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 EWS 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 EWS Protocol Test Suites package. The version of technical specification MS-OXWSITEMID is v20150330 and others are v20121003.

Exchange EWS Protocol technical specifications

|  |  |  |
| --- | --- | --- |
| Technical specification | | Protocol name |
| MS-OXWSATT | [Attachment Handling Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267322) | |
| MS-OXWSBTRF | [Bulk Transfer Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267323) | |
| MS-OXWSCONT | [Contacts Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267324) | |
| MS-OXWSCORE | [Core Items Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267325) | |
| MS-OXWSFOLD | [Folders and Folder Permissions Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267326) | |
| MS-OXWSITEMID | [Web Service Item ID Algorithm](http://go.microsoft.com/fwlink/?LinkID=616012) | |
| MS-OXWSMSG | [E-Mail Message Types Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267327) | |
| MS-OXWSMTGS | [Calendaring Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267328) | |
| MS-OXWSSYNC | [Mailbox Contents Synchronization Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267329) | |
| MS-OXWSTASK | [Tasks Web Service Protocol](http://go.microsoft.com/fwlink/?LinkID=267330) | |

# 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 EWS 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 EWS 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 EWS Protocol Test Suites depend on the Protocol Test Framework (PTF) to derive managed adapters.

# Package design

The Exchange EWS 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 EWS Protocol Test Suites architecture.



**Figure 1: Architecture**

The following outlines the details of the test suites architecture:

**SUT**

The 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
* Microsoft Exchange Server 2019

**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 EWS Protocol Test Suites include one common library, 10 adapters and nine test suites.

* 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.
* Expect MS-OXWSITEMID adapter, all other protocol adapters communicate with SUT through the common library.
* MS-OXWSITEMID protocol adapter implements the Web Service Item Algorithm. All other test suites use the MS-OXWSITEMID protocol adapter to parse and process the Id of an ItemId object in SOAP request and SOAP response.

## Common library

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

### Transport classes

There is one transport class in the common library: ExchangeServiceBinding.

The ExchangeServiceBinding transport class implements SOAP over HTTP and SOAP over HTTPS communications between test suites and SUTs.

### 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.
* Validate certificate used for authentication.
* Other methods which are used by multiple test suites.

### Message structures

Except MS-OXWSITEMID, all protocols are SOAP protocols. Therefore the message structures are used by multiple test suites. These 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, three 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-OXWSITEMID, all other protocol adapters use ExchangeServiceBinding transport class defined in the common library to send and receive messages. MS-OXWSITEMID protocol adapter implements the Web Service Item Algorithm which describes the format of the Id and how to process it. All other test suites use the MS-OXWSITEMID protocol adapter to parse and process the Id of an ItemId object in SOAP request and SOAP response.

### 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. 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 three protocols that have a SUT control adapter in the Exchange EWS Protocol test suites package: MS-OXWSBTRF, MS-OXWSSYNC and MS-OXWSMSG..

## 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.

### MS-OXWSATT

One scenario is designed to verify the server-side, testable requirements in MS-OXWSATT test suite. The following table lists the scenarios designed in the test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_AttachmentProcessing | This scenario is designed to test operations related to the creation, retrieving, and deletion of the attachment. |

### MS-OXWSBTRF

One scenario is designed to verify the server-side, testable requirements in MS-OXWSBTRF test suite. The following table lists the scenarios designed in the test suite.

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_ExportAndUploadItems | This scenario is designed to export items from a mailbox server and upload items to a mailbox server. |

### MS-OXWSCONT

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateGetDeleteContactItem | This scenario is designed to test operations related to creation, retrieving and deletion of the contact items in the server. |
| S02\_UpdateContactItem | This scenario is designed to test operation related to updating of the contact items in the server. |
| S03\_CopyContactItem | This scenario is designed to test operation related to copy of the contact items in the server. |
| S04\_MoveContactItem | This scenario is designed to test operation related to movement of the contact items in the server. |
| S05\_OperateMultipleContactItems | This scenario is designed to test operations related to creation, updating, movement, retrieving, copy and deletion of the multiple contact items in the server. |
| S06\_OperateContactItemWithOptionalElements | This scenario is designed to test operations related to creation, updating, movement, retrieving and copy of the contact items with optional elements in the server. |

### MS-OXWSCORE

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_ManageBaseItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of base items on the server. |
| S02\_ManageContactItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of contact items on the server. |
| S03\_ManageDistributionListsItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of distribution list items on the server. |
| S04\_ManageEmailItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, sending, deletion and mark of email items on the server. |
| S05\_ManageMeetingItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of calendar items on the server. |
| S06\_ManagePostItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of post items on the server. |
| S07\_ManageTaskItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, deletion and mark of task items on the server. |
| S08\_ManageSevenKindsOfItems | This scenario is designed to test operations related to creation, retrieving, updating, movement, copy, and deletion of base, contact, distribution list, email, meeting, post, and task items on the server. |

### MS-OXWSFOLD

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateDeleteFolder | This scenario is designed to test operations related to creation and deletion of folders from a mailbox. |
| S02\_CopyFolder | This scenario is designed to test operations related to the copy of an identified folder. |
| S03\_MoveFolder | This scenario is designed to test operations related to moving folders from a specified parent folder and putting them in another parent folder. |
| S04\_GetFolder | This scenario is designed to test operations related to retrieving folders. |
| S05\_EmptyFolder | This scenario is designed to test operations related to emptying identified folders and deleting subfolders of the specified folder. |
| S06\_UpdateFolder | This scenario is designed to test operations related to the modification properties of an existing folder. |
| S07\_FolderPermission | This scenario is designed to test permission related cases. |
| S08\_OptionalElements | This scenario is designed to test all operations with/without all optional elements. |

### MS-OXWSMSG

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateGetDeleteEmailMessage | This scenario is designed to test operations related to creation, retrieving, and deletion of an email message from the server. |
| S02\_UpdateEmailMessage | This scenario is designed to test operation related to updating an email message on the server. |
| S03\_CopyEmailMessage | This scenario is designed to test operation related to copy of an email message on the server. |
| S04\_MoveEmailMessage | This scenario is designed to test operation related to movement of an email message on the server. |
| S05\_SendEmailMessage | This scenario is designed to test operation related to sending of an email message on the server. |
| S06\_OperateMultipleEmailMessages | This scenario is designed to test operations related to creation, retrieving, updating, copy, movement, sending and deletion of multiple email messages on the server at the same time. |
| S07\_OptionalElementsValidation | This scenario is designed to test operations related to creation, retrieving and deletion of an email message with/without optional elements. |

### MS-OXWSMTGS

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateGetDeleteCalendarRelatedItem | This scenario is designed to test operations related to creation, retrieving and deletion of calendar related items on server. |
| S02\_UpdateCalendarRelatedItem | This scenario is designed to test operation related to updating of calendar related items on server. |
| S03\_CopyCalendarRelatedItem | This scenario is designed to test operation related to copy of calendar related items on server |
| S04\_MoveCalendarRelatedItem | This scenario is designed to test operation related to movement of calendar related items on server. |
| S05\_OperateMultipleCalendarRelatedItems | This scenario is designed to test operations related to creation, retrieving, updating, copy, movement and deletion of multiple calendar related items on server. |

### MS-OXWSSYNC

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_SyncFolderHierarchy | This scenario is designed to test SyncFolderHierarchy operation on the following folders: inbox, calendar, contacts, tasks and search. |
| S02\_SyncFolderItems | The scenario is designed to test SyncFolderItems operation on the following items: MessageType item, MeetingRequestMessageType item, MeetingResponseMessageType item, MeetingCancellationMessageType item, TaskType item, ContactItemType item, PostItemType item, CalendarItemType item, DistributionListType item and ItemType item. |
| S03\_OperateSyncFolderHierarchyOptionalElements | This scenario is designed to test SyncFolderHierarchy operation succeeds to get the synchronization information of five types of folders: inbox, calendar, tasks, contacts, and search regardless of setting the optional elements. |
| S04\_OperateSyncFolderItemsOptionalElements | This scenario is designed to test SyncFolderItems operation succeeds to get the synchronization information of multiple items regardless of setting the optional elements. |

### MS-OXWSTASK

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

|  |  |
| --- | --- |
| Scenario | Description |
| S01\_CreateGetDeleteTaskItem | This scenario is designed to test operations related to creation, retrieving, and deletion of a task item from the server. |
| S02\_UpdateTaskItem | This scenario is designed to test the operation related to updating a task item on the server. |
| S03\_CopyTaskItem | This scenario is designed to test the operation related to copying a task item on the server. |
| S04\_MoveTaskItem | This scenario is designed to test the operation related to moving a task item on the server. |
| S05\_OperateMultipleTaskItems | This scenario is designed to test operation related to operating multiple task items on the server at the same time. |
| S06\_OperateTaskItemWithOptionalElements | This scenario is designed to test operations related to operating a task item with or without optional elements on the server. |