|  |  |
| --- | --- |
| logo_ec_17_colors_300dpi | EUROPEAN COMMISSION  DIGIT  Connecting Europe Facility |

CEF eDelivery Conformance Testing Service

Version 1.11

AS4 User Guide

© European Union, 2020

Reuse of this document is authorised provided the source is acknowledged. The Commission's reuse policy is implemented by Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents.

Document Status:

|  |
| --- |
| **Status** |
| Final |

Document Approver(s):

|  |  |
| --- | --- |
| **Name** | **Role** |
|  |  |
|  |  |
|  |  |

Document Reviewers:

|  |  |
| --- | --- |
| **Name** | **Role** |
| Maarten DANIELS | CEF eDelivery Technical Office |
|  |  |
|  |  |

Summary of Changes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Created by** | **Short Description of Changes** |
| V1.00 |  | Maarten DANIELS | Creation of the document |
| V1.01 | 05/08/2016 | Maarten DANIELS | Updated the introduction chapter to clarify distinction between conformance testing and connectivity testing  Changed sequence of Deliver and Notify in 3.2.1 and 3.6  Added section 4.2 to explain the connectivity testing scenarios into more detail |
| V1.02 | 11/08/2016 | Maarten DANIELS | Added flowcharts to section 4.2 |
| V1.03 | 31/08/2016 | Maarten DANIELS | Updated endpoints from 169.50.130.202 to 13.81.109.44 |
| V1.04 | 22/09/2016 | Maarten DANIELS | Clarified that retry interval must be less than 3 minutes and updated OneWayRetry scenario description |
| V1.05 | 04/10/2016 | Maarten DANIELS | Updated PartyId type in sample messages in section 3.6 after Kerkovi 1.4.1 update |
| V1.06 | 29/01/2018 | Maarten DANIELS | Updated description of Conformance Testing service and Connectivity Testing service in section 1 |
| V1.07 | 01/06/2018 | Maarten DANIELS | Updated document to reflect the changes resulting from the modularization of the eDelivery AS4 profile  Updated references to e-SENS |
| V1.08 | 27/09/2018 | Caroline AEBY | Contact information update |
| V1.09 | 15/11/2018 | Maarten DANIELS | Added section on ENTSOG Usage Profile Addendum Test Assertions  Updated section 3.7 |
| V1.10 | 10/12/2018 | Maarten DANIELS | Added domain names in addition to the IP addresses of the Test Platform endpoints |
| V1.11 | 10/04/2020 | Maarten DANIELS | Updated section 4.5 |

Table of Contents

[Approach and purpose of the document 5](#_Toc532199185)

[1. Introduction 6](#_Toc532199186)

[1.1. Modularization of the eDelivery AS4 profile 7](#_Toc532199187)

[2. Overview 8](#_Toc532199188)

[3. Connecting an Access Point to the test platform 9](#_Toc532199189)

[3.1. Sample Workflow 9](#_Toc532199190)

[3.1.1. Assumptions 9](#_Toc532199191)

[3.1.2. Sequence 9](#_Toc532199192)

[3.2. Access Point C2-C3 PModes 9](#_Toc532199193)

[3.2.1. PartyID - Address Mapping 10](#_Toc532199194)

[3.2.2. PMode Profiles and Service-Action Mappings 10](#_Toc532199195)

[3.2.3. Payload Profiles 11](#_Toc532199196)

[3.3. Backend C1-C2 or C3-C4 PModes 12](#_Toc532199197)

[3.3.1. Overview 12](#_Toc532199198)

[3.3.2. Test Operations 13](#_Toc532199199)

[3.4. Keystores and Certificates 15](#_Toc532199200)

[3.5. Adapters 15](#_Toc532199201)

[3.5.1. Backend Adapters 16](#_Toc532199202)

[3.5.2. Man in the middle (AS4-interceptor) 16](#_Toc532199203)

[3.6. Sample Messages 16](#_Toc532199204)

[3.6.1. Sample Submission from Minder (C1) to C2 17](#_Toc532199205)

[3.6.2. Sample Request Message (from C2 to C3) 17](#_Toc532199206)

[3.6.3. Sample Notification (from C2 to C1) 17](#_Toc532199207)

[3.6.4. Sample Delivery (from C3 to C4) 17](#_Toc532199208)

[3.6.5. Sample Reply Message (from C3 to C2) 18](#_Toc532199209)

[3.7. Check list 18](#_Toc532199210)

[4. Running tests on the test platform 19](#_Toc532199211)

[4.1. Logging in to Minder 19](#_Toc532199212)

[4.2. AS4 Basic Connectivity Tests 20](#_Toc532199213)

[4.2.1. AS4\_basic\_one\_way 20](#_Toc532199214)

[4.2.2. AS4\_basic\_two\_way 21](#_Toc532199215)

[4.2.3. OnewaySignOnly 24](#_Toc532199216)

[4.2.4. OneWayRetry 25](#_Toc532199217)

[4.3. AS4 Common Profile Test Assertions 28](#_Toc532199218)

[4.4. AS4 Four Corner Profile Enhancement Test Assertions 29](#_Toc532199219)

[4.5. AS4 SBDH Profile Enhancement Tests Assertions 30](#_Toc532199220)

[4.6. AS4 Dynamic Receiver Profile Enhancement Test Assertions 31](#_Toc532199221)

[4.7. AS4 Dynamic Sender Profile Enhancement Test Assertions 32](#_Toc532199222)

[4.8. ENTSOG Usage Profile Addendum Test Assertions 33](#_Toc532199223)

[4.8.1. Sample ENTSOG Submission from Minder (C1) to C2 34](#_Toc532199224)

[5. Annex 35](#_Toc532199225)

[6. Contact information 36](#_Toc532199226)

Approach and purpose of the document

The present document is the User Guide of the CEF eDelivery Conformance Testing service. Key content includes an explanation of the components of the CEF eDelivery Conformance Testing service and the actions required from the software providers connecting making use of the service.

Glossary

The key terms used in this Service Offering Description are defined in the CEF Definitions section on the CEF Digital Single Web Portal:

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Definitions>

The key acronyms used in this Service Offering Description are defined in the CEF Glossary on the CEF Digital Single Web Portal:

<https://ec.europa.eu/cefdigital/wiki/pages/viewpage.action?spaceKey=CEFDIGITAL&title=CEF+Glossary>

Acknowledgments

This guide is based on the work of the e-SENS project and the documentation of the Minder Test Platform and Kerkovi bridge by Tubitak in the context of e-SENS.

Introduction

This document describes how software providers can use the Conformance Test Platform. It clarifies the different components of the platform, explains how to connect to the platform and details how to run the test cases against an implementation of a conformant CEF eDelivery component.

Note that CEF e-Delivery offers two different types of testing services:

* Conformance Testing service
* Connectivity Testing service

The goal of the CEF eDelivery Conformance Testing service is to verify that an implementation of the CEF eDelivery Access Point and SMP specifications, a software package either commercial or Open Source, conforms to the relevant CEF eDelivery specifications.

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Conformance+testing>

The goal of the CEF eDelivery Connectivity Testing service is to test if a newly installed AS4 Access Point, conformant with the CEF eDelivery specifications, can successfully communicate with the sample AS4 Access Point hosted by the European Commission or if a newly installed SMP server, conformant with the CEF eDelivery specifications, can respond to sample requests. If successful, these tests confirm that the relevant component is in all likelihood correctly deployed and configured.

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Connectivity+testing>

This document only describes the Conformance Testing service.

Modularization of the eDelivery AS4 profile

As of version 1.13, the eDelivery AS4 profile is defined in a modular structure. This increases the reusability of the specifications in diverse domains, and lowers the implementation burden for the many users that only need a small subset of the existing functionality. It will also make it easier in the future to add new features or to combine eDelivery with other technologies.

The eDelivery AS4 profile is now reorganized into a Common Profile and a number of optional, independent add-on Profile Enhancements. The Common Profile covers all selected technical and security-related parameters and is independent of specific domains, topologies and discovery and security infrastructures. The Enhancement provide additional optional extensions.

An implementation performing Conformance Testing can choose which optional Profile Enhancements to support in addition to the mandatory "Common Profile". An implementation listed as a Conformant Solution is only marked conformant to the optional Profile Enhancements for which all mandatory tests have been successfully completed.

The specific requirements and instructions for each specific Profile Enhancement are explained in chapter 4 in this document.

Implementations that were conformant prior to the modularization of the AS4 profile will remain eDelivery AS4 conformant against the "Common Profile" and "Four Corner Profile Enhancement". Conformance to the other Profile Enhancements can be achieved by successfully passing all mandatory tests in the test group of the specific Profile Enhancement.

Overview

The Conformance Test Platform consists of two main components.

* Minder testbed: a generic online programmable flow control engine that provides the capability of interconnecting different systems in one node and enabling architects to perform complex communication operations on those interconnected systems.
* Kerkovi bridge: the generic AS4 Conformance & Interoperability bridge that achieves connectivity between the testing system (Minder) and the AS4 conformant Access Points.

Kerkovi consists of three adapters (two backend simulators and an AS4 interceptor). The backend adapters are capable of sending AS4 messages as backend submission messages to the AS4 Access Points. They can also receive delivery, submission result and notification messages.

Kerkovi is responsible of tracking the addresses of the Access Points with respect to the party ID's and forwarding of messages with respect to the addresses it resolves from the party ID's.

Connecting an Access Point to the test platform

This section describes how to connect an Access Point to Minder via the Kerkovi bridge. Since some test cases test the Access Point as a sender and some test cases test the Access Point as a receiver, two instances of the same Access Point implementation are required to connect to the test platform.

Sample Workflow

A sample message flow that includes the journey of a message from Corner1 to Corner 4 is given below.

Assumptions

C1: Producer = Minder.

C2: Sending MSH, which sends messages to Corner 3 (C3). C2 is a first instance of the system under test (SUT).

C3: Receiving MSH, which receives messages and sends back receipts (i.e. in a ONE\_WAY MEP scenario). C3 is a second instance of the system under test (SUT).

C4: Consumer = Minder.

Sequence

1. *Submit:* C1 sends a message to C2
2. C1 receives a receipt from C2 containing details of the Producer message - the receipt back to C1 is optional and may just be an HTTP Response 20x
3. Send*:* C2 constructs a message based on the message received from C1 and sends this to C3
4. C2 receives a receipt (or error) signal from C3 containing details of the C2 message
5. *Deliver:* C3 constructs a message based on the message it received, and delivers it to C4.
6. *Notify:* C2 constructs a message detailing the result of the message from C2 to C3 and sends this to C1

Access Point C2-C3 PModes

The Conformance Testing environment requires specific PMode settings for the tests to be executed properly. These PMode settings[[1]](#footnote-1) are based on the eDelivery AS4 profile.

PartyID - Address Mapping

Since the AS4 Conformance test architecture requires the interception of the message flow between C2 and C3, all the participating Access Points must send the messages directly to the interceptor and not the actual target Access Point. The Interceptor address is <https://13.81.109.44:15000/as4Interceptor> (<https://edelconf.westeurope.cloudapp.azure.com:15000/as4Interceptor>) or <http://13.81.109.44:15001/as4Interceptor> (<http://edelconf.westeurope.cloudapp.azure.com:15001/as4Interceptor>). If the message should be forwarded to the receiver side with respect to the test scenario, then Minder will do it.

PMode Profiles and Service-Action Mappings

The Access Points should be able to identify and select the correct PMode profile below, with respect to the Service and Action parameters that they receive from the backend.  It is up to the Access Point to map its PMode by the Service or the Action, which are submitted as message properties to the Access Point via the AS4 backend.

All the other PMode parameters are the same for as the eDelivery AS4 profile PModes.

The [eDelivery AS4 test assertions](https://169.50.130.202:15000/TestAssertions_eSENS_AS4_Draft6.pdf)[[2]](#footnote-2) require five different profiles based on the eDelivery AS4 profile PModes. They are provided below.

|  |
| --- |
| 1. SIMPLE\_ONEWAY    1. **MEP**: One way - push    2. **Compress**: Yes    3. **Retry**: None    4. **Sign**: Yes    5. **Encrypt**: Yes    6. **Service**: SRV\_SIMPLE\_ONEWAY    7. **Action**: ACT\_SIMPLE\_ONEWAY |
| 1. SIMPLE\_TWOWAY    1. **MEP**: Two way push-and-push    2. **Compress**: Yes    3. **Retry**: None    4. **Sign**: Yes    5. **Encrypt**: Yes    6. **Service**: SRV\_SIMPLE\_TWOWAY    7. **Action**: ACT\_SIMPLE\_TWOWAY |
| 1. ONEWAY\_RETRY    1. **MEP**: One way - push    2. **Compress**: Yes    3. **Retry**: 5 (the interval between retries must be less than 3 minutes)    4. **Sign**: Yes    5. **Encrypt**: Yes    6. **Service**: SRV\_ONEWAY\_RETRY    7. **Action**: ACT\_ONEWAY\_RETRY |
| 1. ONEWAY\_ONLY\_SIGN    1. **MEP**: One way - push    2. **Compress**: Yes    3. **Retry**: None    4. **Sign**: Yes    5. **Encrypt**: No    6. **Service**: SRV\_ONEWAY\_SIGNONLY    7. **Action**: ACT\_ONEWAY\_SIGNONLY |
| 1. PING    1. **MEP**: One way - push    2. **Compress**: Yes    3. **Retry**: None    4. **Sign**: Yes    5. **Encrypt**: Yes    6. **Service**: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/service    7. **Action**: http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/test |

Payload Profiles

The test cases that are created for the test assertions submit a combination of types of payloads with the following information.

|  |
| --- |
| 1. xmlpayload@minder 2. **ContentId**=xmlpayload@minder 3. **MimeType**=application/xml 4. **Required**=yes 5. **Character Set**=utf-8 (optional) |
| 1. xmlpayload2@minder 2. **ContentId**=xmlpayload2@minder 3. **MimeType**=application/xml 4. **Required**=No 5. **Character Set**=utf-8 (optional) |
| 1. custompayload@minder 2. **ContentId**=custompayload@minder 3. **MimeType**=application/octet-stream 4. **Required**=No |

Backend C1-C2 or C3-C4 PModes

This section provides the PMode definitions to be used in operations between Minder and the system under test (SUT) where Minder acts as Corner 1 (C1) or Corner (4) and the SUT acts as Corner 2 (C2) or Corner 3 (C3) in the four corner model.

This is based on the Messaging Model diagram shown below:

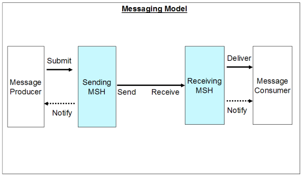


Figure 1: Messaging Model diagram

For the PMode definition considered here Minder will act as the Message Producer / Consumer and the SUT is either the Sending or Receiving MSH. Therefore PModes need to be defined for the Submit, Deliver and Notify operations. The PModes that define the message exchanges between the MSHs are different and are detailed in 3.2 Access Point C2-C3 PModes.

Overview

All PModes use the Push MEP binding. Because the submit operation may require a response to inform the Message Producer (Minder) on the eb:MessageId used in the exchange between the MSHs it uses a Two-Way MEP. Both the Deliver and Notify operation use a One-Way MEP.

The PartyIds to use for Minder and the SUT have to be agreed upon between the Conformance Testing team and the software provider of the SUT.

The proposed values for Role are as follows

* http://www.esens.eu/as4/conformancetest/testdriver for Minder
* http://www.esens.eu/as4/conformancetest/sut for the system under test.

AgreementRef is not used and must not be included in the messages.

The Service value to use in all PModes is "http://www.esens.eu/as4/conformancetest" with no type attribute.

No security and reliability is used between the SUT and Minder.

Test Operations

The operations associated with the three PModes are 'Submit', 'Deliver' and 'Notify'. Each operation message between Minder and the SUT includes a set of Message Properties defining the necessary instructions and properties for the test operation as defined below.

Submit

Instruction from Minder to C2 or C3 to submit a message to the opposite corner (C3 or C2 respectively). Because Minder needs to know the ebMS MessageId that the SUT uses for the exchange with the other corner the submit operation consists of two legs with Minder always being the Initiator.

The request leg of the submit is the actual submission from Minder to the SUT. The value to use for Action is Submit.

* Initiator = minder
* Action = Submit

In the eb:MessageProperties element the meta-data required for submitting the message from Minder to the SUT must be supplied.

It is assumed that the From PartyId to use in the message to be sent by the MSH is fixed and does not need to be provided by Minder. The originalSender and finalRecipient properties must also be included in the eb:MessageProperties  of the actual message sent by the MSH to the other MSH.

Based on these message properties the SUT must be able to find the PMode to use for the message exchange with the other MSH. The PModes that define the message exchanges between the MSHs should therefore be sufficiently different to allow unambiguous selection of the PMode.

|  |  |
| --- | --- |
| Property Name | Required? |
| MessageId | N |
| ConversationId | Y |
| RefToMessageId | N |
| ToPartyId | Y |
| ToPartyRole | Y |
| Service | Y |
| ServiceType | N |
| Action | Y |
| originalSender | Y |
| finalRecipient | Y |

The payloads included in the submit message (from Minder to SUT) and associated meta-data provided in eb:PayloadInfo must also be included with the message sent by the SUT.

Deliver

The deliver operation is between the opposite inner corner SUT and Minder. I.e. C3 or C2 where C2 or C3 received the corresponding Minder submit operation. Initiator and action are set as follows:

* Initiator = SUT
* Action = Deliver

The deliver operation is the reverse of submit, so the meta-data can be transferred from the SUT to Minder in the same way as in the submit operation.

Although the deliver operation is abstract, i.e. has no formal specification, it is assumed that implementations are able to supply the specified meta-data on delivery.

Notify

The notify operation is from the corner which received a Minder submit operation or in case of receipt/error signal from the opposite corner. The aim of this operation is to inform the Producer about the future status of a message previously submitted to the SUT. Therefore, the SUT should inform the Producer in case a message is successfully sent to the opposite corner, or an error occurs during the delivery operation. In both error and success cases, the Producer must be able to correlate the notification message with the previously submitted message. Therefore, the RefToMessageId field is required in both error and success cases.

Additionally, a SUT does not have to notify the Producer about an error that is not related to a previously submitted message.

* Initiator = SUT
* Action = Notify

The following message properties are defined to indicate the corresponding message for which the Signal was received and the Signal type (Receipt or Error).

|  |  |  |
| --- | --- | --- |
| Property name< | Required? | Notes |
| RefToMessageId | Y/O | Required for Receipts, Optional for Errors |
| SignalType | Y | Receipt or Error |
| ErrorCode | N/Y | EBMS:xxxx for Errors only |
| ShortDescription | O |  |
| Description | O |  |

The RefToMessageId property is optional in the case of an ebMS Error Signal. It must be set to the eb:RefToMessageId of the signal which is the eb:MessageId of the original message that was sent by the SUT to the opposite corner if included.

The SignalType property must contain either Receipt or Error. In addition to the message properties the message may include information on the received signal as a payload.

The ErrorCode property should reflect the value of ebMS error code in case of a fault message. For a Receipt notification type, this value may be empty.

The ShortDescription and Description properties should reflect the values of the same name (i.e. eb:Error@shortDescription and eb:Error/eb:Description) in case of a fault message.

Keystores and Certificates

The keystores and certificates that are used for communication between C2 and C3 have to be known by Minder. This is because some test assertions require decryption and/or even modification of the messages. Upon registration to the Conformance Test Service, you will receive a certificate/key that is shared with the Conformance Testing team.

Adapters

Kerkovi provides three adapters (i.e. actors in GITB terms) to create a bridge between Minder and the conformance testing environment.

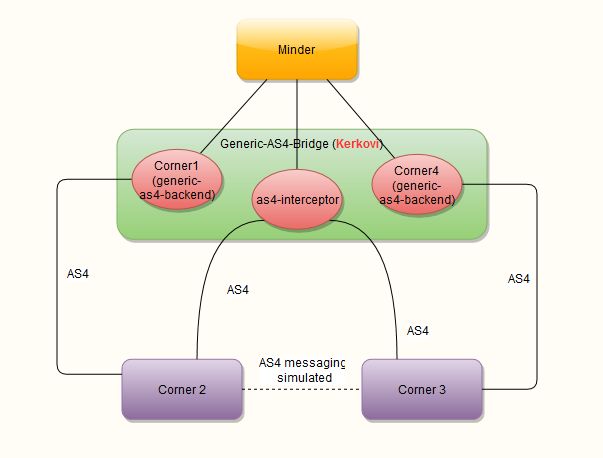


Figure 2: Adapters

Backend Adapters

There are two backend adapters (i.e. corner1 and corner4). They send messages to and receive messages from the AS4 Access Points as described in the PMode documentation. A generic AS4 backend sends a *Submit.*

|  |
| --- |
| Corner 1 (generic AS4 backend)  Plays the role of *Corner 1* (e.g. the sending backend) and submits messages to the *backend* of the AS4 compliant Access Point. It also receives messages that have been relayed to Corner 2 (from Corner 3).  Endpoint address:  <https://13.81.109.44:15000/corner1> (<https://edelconf.westeurope.cloudapp.azure.com:15000/corner1>) or  <http://13.81.109.44:15001/corner1> (<http://edelconf.westeurope.cloudapp.azure.com:15001/corner1>) |
| Corner 4 (generic AS4 backend)  Plays the role of *Corner 4* (e.g. the receiving backend) and submits messages to the *backend* of the AS4 compliant Access Point. It also receives messages that have been relayed to Corner 3.  Endpoint address:  <https://13.81.109.44:15000/corner4> (<https://edelconf.westeurope.cloudapp.azure.com:15000/corner4>) or <http://13.81.109.44:15001/corner4> (<http://edelconf.westeurope.cloudapp.azure.com:15001/corner4>) |

Man in the middle (AS4-interceptor)

|  |
| --- |
| Intercepts the message flow between Corner2  and Corner3  and depending on the Proxy settings, forwards the message either to Minder or its target (resolved from the TO\_PARTY\_ID)  Endpoint address:  <https://13.81.109.44:15000/as4Interceptor> (<https://edelconf.westeurope.cloudapp.azure.com:15000/as4Interceptor>) or <http://13.81.109.44:15001/as4Interceptor> (<http://edelconf.westeurope.cloudapp.azure.com:15001/as4Interceptor>) |

Sample Messages

In this section we provide sample AS4 messages for a TWO\_WAY MEP that may take place in a sample conformance scenario. We have also depicted the multipart mime message format for the AS4 messages that carry attachments. These samples may be used as a starting point for a test run.

The ebMS ID's of the messages (MessageID) are given below with respect to their occurrence in the Sample Workflow.

|  |  |  |
| --- | --- | --- |
| Name | Value | Step |
| Submission From C1 to C2: | submission@corner1 | Step 1 |
| AS4 Message from C2 to C3 (request leg): | as4message@corner2 | Step 3 |
| Deliver from C3 to C4: | deliver@corner3 | Step 5 |
| Notification from C2 to C1: | notification@corner2 | Step 6 |
| AS4 Message from C3 to C2 (reply leg): | as4message@corner3 | Not given in steps |

No information about the receipts is provided as they are optional. If you want to respond with receipts, refer to the [AS4 specification](http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/os/AS4-profile-v1.0-os.html).

Sample Submission from Minder (C1) to C2



Sample Request Message (from C2 to C3)



Sample Notification (from C2 to C1)



Sample Delivery (from C3 to C4)



Sample Reply Message (from C3 to C2)



Check list

The checklist below is a good summary of what must be done before starting a Conformance Test session via Minder and Kerkovi.

1. Register for the Conformance Testing Service according to the guidelines in the Conformance Testing Service Offering Description[[3]](#footnote-3).
2. Test platform integration
   1. Study the PModes for the communication between C1 - C2 and C3 - C4.
   2. Implement the Submit action
   3. Implement the Notify action
   4. Implement the Deliver action
3. Preparing for the actual the test cases:
   1. Check the Test Assertions document to see if your implementation covers all the tests
   2. Study the PModes for the communication between C2 - C3
   3. Take a look at the sample messages
4. Configure your solution and run the basic connectivity tests

Running tests on the test platform

Logging in to Minder

Use a browser to navigate to the Minder Test Platform hosted by CEF eDelivery[[4]](#footnote-4) and enter the username/password combination you have received upon registration to the Conformance Test Service.

Click the "Inventory" tab and select one of the following test groups:

* AS4 Basic Connectivity Tests[[5]](#footnote-5)
* AS4 Common Profile Test Assertions
* AS4 Four Corner Profile Enhancement Test Assertions
* AS4 SBDH Profile Enhancement Tests Assertions
* AS4 Dynamic Receiver Profile Enhancement Test Assertions
* AS4 Dynamic Sender Profile Enhancement Test Assertions

If a group has been selected, all the test assertions belonging to this group are displayed.

Click the tab "Test Suites" to select the test suite that starts with the name of your software implementation.

The screen that is displayed shows all the test assertions that can be run against your instances.

Use the button "Run All Tdls" to run all test assertions in sequence or Use the button "Run Selected Tdls" to run only a specific set of test assertions in sequence.

The test platform will now ask the visibility settings of the test run.

* Public: All registered users can see the details of the test run.
* Protected: All Test Designers can see the details of the test run.
* Private: Only the owner can see the details of the test run.

The recommended setting is Protected. This will prevent that other software implementers that use the platform can see each other's test results while still allowing the Test Designers and administrators of the Test Platform to monitor the results of the tests.

The progress of the test execution can be followed by selecting the "Test Monitor" tab at the top of the page or by refreshing the test suite page.

Once a test run is finished the "Batch Job Details" are available at the bottom of the test suite page and a test report can be generated by clicking the buttons "Report All Runs" or "Report Selected Runs".

AS4 Basic Connectivity Tests

Before executing the test assertions that are relevant for passing the actual conformance tests from the mandatory Common Profile or the optional Profile Enhancement modules (listed their respective test groups), it is recommended that the tests listed in the AS4 Basic Connectivity Tests group are executed.

The AS4 Basic Connectivity Tests are a group of tests to verify the connectivity to the Minder Test Platform only. They are not related to the CEF e-Delivery Connectivity Testing service described in <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Connectivity+testing>.

Since they are not part of the Test Assertion document, the test cases from the AS4 Basic Connectivity Tests group are explained below.

AS4\_basic\_one\_way

1. C1 sends a Submit message to C2. For details, see section 3.3.2.1 Submit.
2. C2 sends back an AS4 Receipt to C1 (unsigned since this is covered by the PMode settings between C1 and C2).
3. C2 sends the payload to C3 via the as4Interceptor (encrypted, signed, compressed, for details see SIMPLE\_ONEWAY).
4. C3 sends back an AS4 Receipt to C2 via the as4Interceptor (constructed according to the applicable PMode settings between C2 and C3).
5. C3 sends a Deliver message to C4 detailing the result of the message from C2 to C3. For details, see section 3.3.2.2 Deliver.
6. C2 sends a Notify message to C1 detailing the result of the message from C2 to C3. For details, see section 3.3.2.3 Notify.

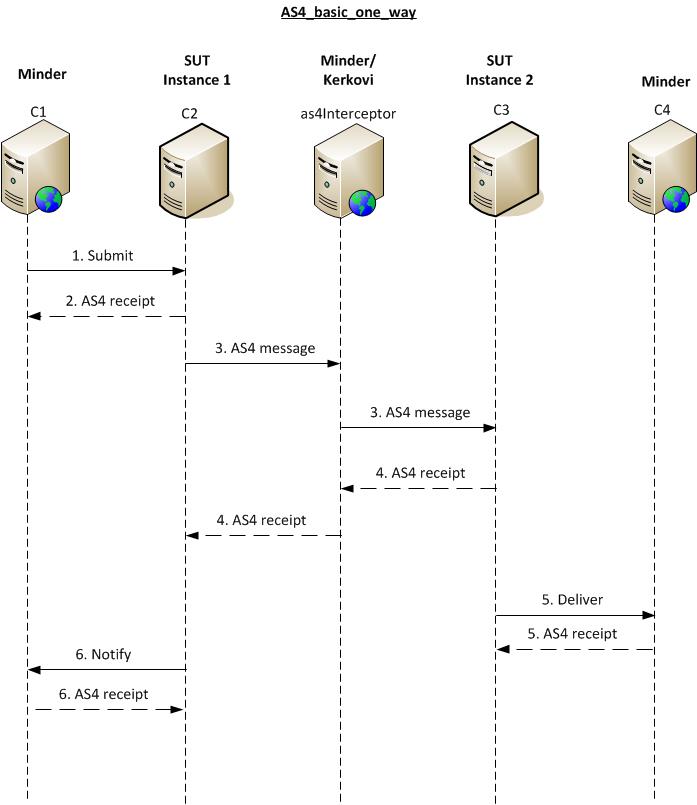


Figure 3: AS4\_basic\_one\_way

AS4\_basic\_two\_way

1. C1 sends a Submit message to C2. For details, see section 3.3.2.1 Submit.
2. C2 sends back an AS4 Receipt to C1 (unsigned since this is covered by the PMode settings between C1 and C2).
3. C2 sends the payload to C3 via the as4Interceptor (encrypted, signed, compressed, for details see SIMPLE\_TWOWAY).
4. C3 sends back an AS4 Receipt to C2 via the as4Interceptor (constructed according to the applicable PMode settings between C2 and C3).
5. C3 sends a Deliver message to C4 detailing the result of the message from C2 to C3. For details, see section 3.3.2.2 Deliver.
6. C2 sends a Notify message to C1 detailing the result of the message from C2 to C3. For details, see section 3.3.2.3 Notify.
7. C4 sends a Submit message to C3. For details, see section 3.3.2.1 Submit.
8. C3 sends back an AS4 Receipt to C4 (unsigned since this is covered by the PMode settings between C3 and C4).
9. C3 sends the payload to C2 via the as4Interceptor (encrypted, signed, compressed).
10. C2 sends back an AS4 Receipt to C3 via the as4Interceptor (constructed according to the applicable PMode settings between C3 and C2).
11. C2 sends a Deliver message to C1 detailing the result of the message from C3 to C2. For details, see section 3.3.2.2 Deliver.
12. C3 sends a Notify message to C4 detailing the result of the message from C3 to C2. For details, see section 3.3.2.3 Notify.

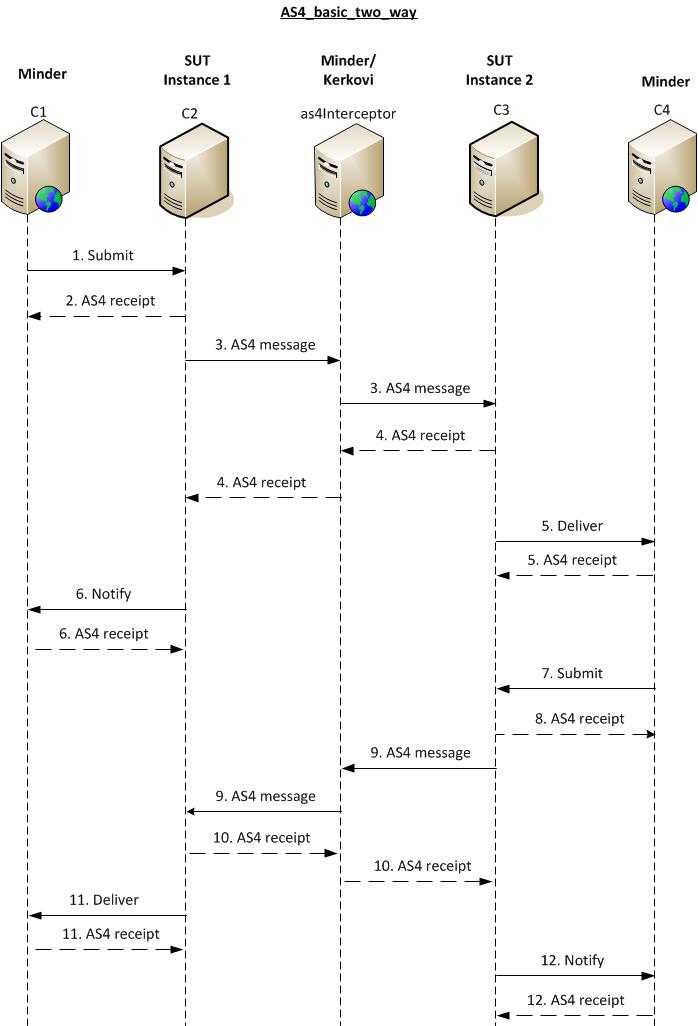


Figure 4: AS4\_basic\_two\_way

OnewaySignOnly

1. C1 sends a Submit message to C2. For details, see section 3.3.2.1 Submit.
2. C2 sends back an AS4 Receipt to C1 (unsigned since this is covered by the PMode settings between C1 and C2).
3. C2 sends the payload to C3 via the as4Interceptor (unencrypted, signed, compressed, for details see ONEWAY\_ONLY\_SIGN).
4. C3 sends back an AS4 Receipt to C2 via the as4Interceptor (constructed according to the applicable PMode settings between C2 and C3).
5. C3 sends a Deliver message to C4 detailing the result of the message from C2 to C3. For details, see section 3.3.2.2 Deliver.
6. C2 sends a Notify message to C1 detailing the result of the message from C2 to C3. For details, see section 3.3.2.3 Notify.

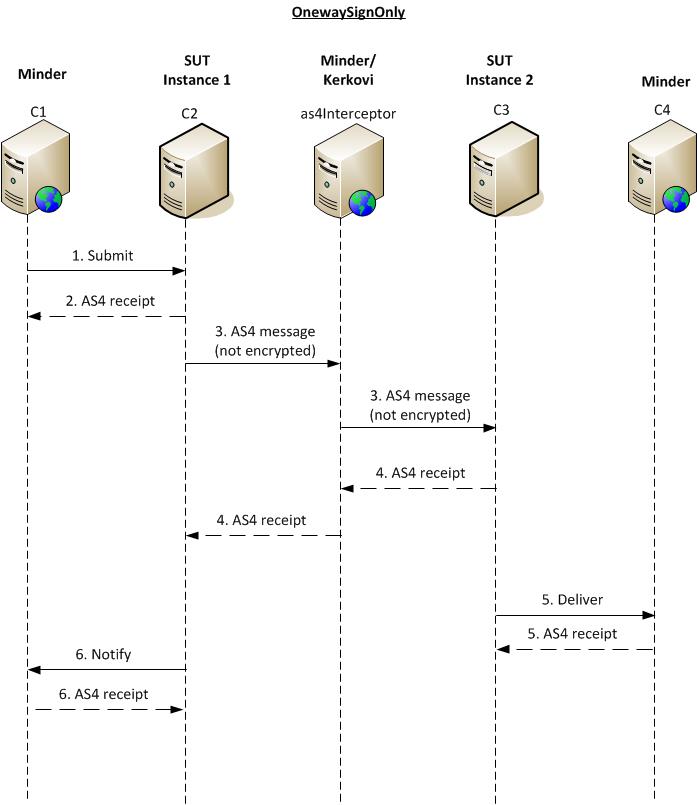


Figure 5: OnewaySignOnly

OneWayRetry

1. C1 sends a Submit message to C2. For details, see section 3.3.2.1 Submit.
2. C2 sends back an AS4 Receipt to C1 (unsigned since this is covered by the PMode settings between C1 and C2).
3. C2 sends the payload to C3 via the as4Interceptor (encrypted, signed, compressed, for details see ONEWAY\_RETRY).
4. For four times, instead of sending the message to C3, the as4Interceptor sends a null reply to C2 and waits for C2 to resend the message to C3 via the as4Interceptor (note that the test platform only waits for 3 minutes, so the retry settings must be configured to resend the message within this time limit). Only the next time, the as4Interceptor actually sends the message to C3.
5. C3 sends back an AS4 Receipt to C2 via the as4Interceptor (constructed according to the applicable PMode settings between C2 and C3).
6. C3 sends a Deliver message to C4 detailing the result of the message from C2 to C3. For details, see section 3.3.2.2 Deliver.
7. C2 sends a Notify message to C1 detailing the result of the message from C2 to C3. For details, see section 3.3.2.3 Notify.

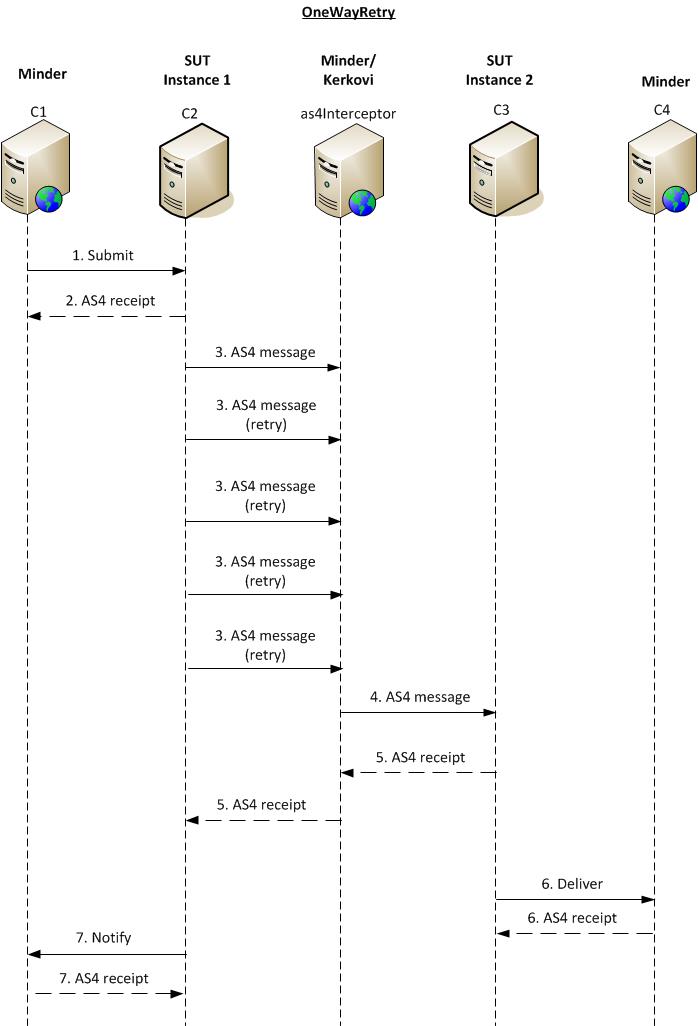


Figure 6: OneWayRetry

AS4 Common Profile Test Assertions

The eDelivery AS4 Common Profile profiles the OASIS ebMS3 and AS4 specifications. More specifically, it selects the AS4 ebHandler Conformance Profile, a specific conformance profile defined in the AS4 specification, and makes a selection of AS4 Advanced Features. The selected Conformance Profile and selected Advanced Features are profiled further for increased consistency, ease of configuration and up to date security. Finally, this Common Profile provides a common AS4 Usage Profile which is mandatory to be supported by all conformant eDelivery AS4 solutions. Additional functionality can be added in optional Profile Enhancements.

This is a mandatory module and all implementations are required to execute this group of Test Assertions.

No specific additional instructions or requirements are needed for this group of Test Assertions.

AS4 Four Corner Profile Enhancement Test Assertions

The Four Corner Topology Profile Enhancement is an optional module to the eDelivery AS4 specifications which allows messages to be exchanged by Access Points on behalf of other parties instead of only using point-to-point message exchanges. In this so-called four corner topology, from an end-to-end perspective, there are four rather than two parties involved in the message exchange. Two parties are the original sender and final recipient parties. The other two parties are Access Points that route messages from the original sender to the final recipient and reverse route response messages.

This is an optional module and all implementations can choose whether to execute this group of Test Assertions.

No specific additional instructions or requirements are needed for this group of Test Assertions.

AS4 SBDH Profile Enhancement Tests Assertions

The SBDH Profile Enhancement is an optional module to the eDelivery AS4 specifications which allows parties to use the eDelivery AS4 Common Profile in conjunction with the UN/CEFACT Standard Business Document Header (SBDH). SBDH is a standard XML format that encodes common message metadata, such as identification of sender and receiver, the type of the payload and the business scope, business process, business transaction, agreement, and business quality-of-service. SBDH is widely adopted in various domains.

This is an optional module and all implementations can choose whether to execute this group of Test Assertions.

No specific additional instructions or requirements are needed for this group of Test Assertions.

AS4 Dynamic Receiver Profile Enhancement Test Assertions

The Dynamic Receiver Profile Enhancement is an optional module to the eDelivery AS4 specifications which allows parties to configure their AS4 MSH to receive user messages from Sender parties that have not been registered in the Receiving MSH and for which the party identifier and signing certificate have not been pre-shared between Sender and Receiver.

This is an optional module and all implementations can choose whether to execute this group of Test Assertions.

The implementation of the test cases in this module is currently ongoing and the specific instructions will be updated when the module is made available on the test platform.

AS4 Dynamic Sender Profile Enhancement Test Assertions

The Dynamic Sender Profile Enhancement is an optional module to the eDelivery AS4 specifications which allows parties to configure their AS4 MSH to send user messages to Receiver parties that have not been pre-configured. For these parties, the party identifier and AS4 protocol parameters (such as the party's encryption certificate and the address of its AS4 server endpoint) are not registered in the Sending MSH. The Profile Enhancement provides a mechanism by which P-Modes can be created dynamically and deployed on an ad hoc basis, by instantiating templates using additional parameters supplied by the Producer and data retrieved using queries on a discovery infrastructure.

This is an optional module and all implementations can choose whether to execute this group of Test Assertions.

The implementation of the test cases in this module is currently ongoing and the specific instructions will be updated when the module is made available on the test platform.

ENTSOG Usage Profile Addendum Test Assertions

The ENTSOG Test Assertions provide optional and additional coverage for the ENTSOG AS4 Usage Profile. An AS4 implementation that, in addition to passing the eDelivery AS4 Common Profile tests, successfully passes these additional tests, provides a more complete implementation for the ENTSOG AS4 profile.

This is an optional module and all implementations can choose whether to execute this group of Test Assertions.

The test cases in this module require specific instructions that a SUT needs to implement in addition to the generic settings specified above in this document.

In addition to the PMode profiles identified in section 3.2, the following PMode profile needs to be supported:

|  |
| --- |
| 1. ENTSOG    1. **MEP**: One way - push    2. **Compress**: Yes    3. **Retry**: None (Note that support for Retry is mandatory in the ENTSOG profile, but it is not used in this test case module to keep tests simple. Support for Retry is already covered in the AS4 Common Profile Test Assertions).    4. **Sign**: Yes    5. **Encrypt**: Yes    6. **Service**: SRV\_ENTSOG    7. **Action**: ACT\_ENTSOG |

This means that the Submit messages from C1 to C2 will include the following service and action message properties: SRV\_ENTSOG and ACT\_ENTSOG.

Based on this PMode profile, an ENTSOG conformant SUT must be able to set the following values in the message between C2 and C3, without being specifically instructed by the Submit message of C1.

1. The mpc attribute may be omitted from the message, but if it is present, it must be set to <http://docs.oasis-open.org/ebxmlmsg/ebms/v3.0/ns/core/200704/defaultMPC>
2. The service type attribute must be set to <http://edigas.org/service>
3. The service value must be taken from an EDIG@S process area code list (e.g. A01, A02, …).
4. The action value must be set to <http://docs.oasis-open.org/ebxml-msg/as4/200902/action>
5. For the business document part a Property SHOULD be included in the PartProperties with a name EDIGASDocumentType set to the same value as the top level type element in the EDIG@S XML document, which is of type DocumentType. This means that the EDIGASDocumentType must be derived from the payload from the Submit message of C1.
6. Contrary to the other modules defined in this document where AgreementRef must not be used, the ENTSOG test assertions mandate the AgreementRef element to be present in the message between C2 and C3.

Other message property values to be used in the message between C2 and C3 are specified in the Submit message from C1 are as following.

1. A message property with the name FromPartyIdType defines the PartyInfo/From/PartyId type attribute. This means that if this message property is set, the default type value urn:oasis:names:tc:ebcore:partyid-type:unregistered needs to be overridden.
2. A message property with the name ToPartyIdType defines the PartyInfo/To/PartyId type attribute. This means that if this message property is set, the default type value urn:oasis:names:tc:ebcore:partyid-type:unregistered needs to be overridden.
3. As the Parties to be used in the ENTSOG network must be using an appropriate EIC Code, the message properties FromPartyId and ToPartyId of the Submit message of C1 contain different PartyInfo/From/PartyId and PartyInfo/To/PartyId values to be used in the AS4 message between C2 and C3 (different from the C1/C2 and C3/C4 transactions).

Sample ENTSOG Submission from Minder (C1) to C2



Annex

Not applicable

Contact information

|  |
| --- |
| CEF Support Team  By email: CEF-EDELIVERY-SUPPORT@ec.europa.eu  Support Service: 8am to 6pm (Normal EC working Days) |

1. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4 [↑](#footnote-ref-1)
2. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Conformance+testing [↑](#footnote-ref-2)
3. https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Conformance+testing [↑](#footnote-ref-3)
4. <https://edelconf.westeurope.cloudapp.azure.com:9001> [↑](#footnote-ref-4)
5. These AS4 Basic Connectivity Tests are a group of tests to verify the connectivity to the Minder Test Platform only. They are not related to the CEF e-Delivery Connectivity Testing service described in <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+Connectivity+testing> [↑](#footnote-ref-5)