Revenue
Cáin agus Custaim na hÉireann
Irish Tax and Customs

# eCustoms

# PIT Connectivity Testing Scenarios

*The information in this document is provided as a guide only and is not professional advice, including legal advice. It should not be assumed that the guidance is comprehensive or that it provides a definitive answer in every case.*

# Contents

| Version Control | | |
|---|---|---|
| **Version** | **Date** | **Change** |
| 1.0 | 22/07/2025 | Initial document |
| | | |
| | | |
| | | |
| | | |

## Audience

This document is for any software provider who wish to test B2G software interacting with the following Customs and Excise systems:

- AIS – Automated Import System (UCC)
- AES – Automated Export System (UCC)
- NCTS – New Customs Transit System (UCC)
- RORO – Roll On Roll Off system
- EDE – Excise Duty Entry
- EMCS – Excise Movement Control System

## Document context

This document describes the connectivity test scenarios that are supported by Revenue in a dedicated Public Interface Testing environment. This document is designed to be read in conjunction with the SOAP and REST integration guides as well as the rest of the Revenue Commissioners' documentation suite including the relevant technical documents.

This document partially supersedes previous *"eCustoms UCC Programme PIT Connectivity and Sample Declaration Test Scenarios"*.

It does not describe presented in the *"eCustoms UCC Programme PIT Connectivity and Sample Declaration Test Scenarios"* document functional test scenarios (sample declarations) specific to each of the Customs systems. For these, please refer to PIT documentation specific to the individual system, as listed in the Document References section.

# Document References

| Reference |
|---|
| 1. *Customs and Excise Web Services Common Specification* |
| 2. *Customs & Excise REST Web Service Integration Guide* |
| 3. *Customs & Excise SOAP Web Service Integration Guide* |
| 4. *Customs & Excise AES PIT Specification* |
| 5. *Customs & Excise AIS PIT Specification* |
| 6. *Customs & Excise NCTS PIT Specification* |

# Abbreviations Used in This Document

| Abbreviation | Description |
|---|---|
| AES | Automated Export System |
| AIS | Automated Import System |
| C&E | Customs and Excise |
| EDE | Excise Duty Entry |
| EMCS | Excise Movement Control System |
| NCTS | New Computerised Transit System |
| PIT | Public Interface Testing |
| RORO | Roll-On Roll-Off system |
| ROS | Revenue Online Service |
| UCC | Union Customs Code |

# 1 Introduction

This document enables eCustoms software developers create/update their software packages to be compatible with current Revenue Customs and Excise request security and digital signatures.

This document provides test scenario examples that are supported by Revenue in its PIT environment to enable software developers validate and verify their connection with the PIT environment.

# 2 The Scope

The document specifically details the test scenarios encompassing the Connectivity Testing phase for PIT. It is strongly recommended that that these scenarios are successfully executed before executing your own test scenarios.

# 3 Live- and Development-Level PIT Environments

Revenue provides a Public Interface Test (PIT) facility where developers may access Schemas and Notes, Web Service specifications, troubleshooting information and technical FAQs.

The PIT facility is provided at two different software levels: Live and Development.

## 3.1 Live-Level PIT

Live Level uses the same software version as the ROS production code. This version should be used if you are developing software that will be compatible with forms and services already released live by Revenue.

| **Live-Level PIT URL:** | | |
|---|---|---|
| https://**softwaretest.ros.ie**/customs/webservice/v1/rest<br>https://**softwaretest.ros.ie**/customs/webservice/v1/soap | | |
| **Customs & Excise Services Available:** | | |
| **Service** | **Enpoint** | **Notes:** |
| Handshake | /handshake | General handshake to verify connectivity and correctness of digital signature. |
| Transaction ID | /transactionID | Obtain transaction ID tokens for Reliable Messaging. |

| | | |
|---|---|---|
| AIS Submit | /aisSubmit | Automated Import System.<br>**NOTE: Only web service URL is provided without actual backend AIS functionality. This will be added in the future.** |
| AES Submit | /aesSubmit | Automated Export System.<br>**NOTE: Only web service URL is provided without actual backend AES functionality. This will be added in the future.** |
| NCTS Submit | /nctsSubmit | New Computerised Transit System.<br>**NOTE: Only web service URL is provided without actual backend NCTS functionality. This will be added in the future.** |
| EDE Submit | /edeSubmit | Excise Duty Entry. |
| EMCS Submit | /emcsSubmit | Excise Movement Control System. |
| Mailbox Collect | /mailboxCollect | Collect Customs and Excise responses. |
| Mailbox Acknowledge | /mailboxAcknowledge | Acknowledge the collection of Customs and Excise responses. |
| Export Release Verification | /export/releaseVerification | Submit ERV requests for Automated Export System.<br>**NOTE: Only web service URL is provided without actual backend AES ERV functionality. This will be added in the future.** |

## 3.2 Development-Level PIT

Development Level version may include new features and forms that are still under development, and which have not as yet been released to production. You should use this version only if you are developing software that will be compatible with a future release of ROS. For all other testing we would recommend using the Live Level version.

| **Development-Level PIT URLs:** | | |
|---|---|---|
| https://**softwaretestnextversion.ros.ie**/customs/webservice/v1/rest<br>https://**softwaretestnextversion.ros.ie**/customs/webservice/v1/soap | | |
| **Customs & Excise Services Available:** | | |
| **Service** | **Enpoint** | **Notes:** |
| Handshake | /handshake | General handshake to verify connectivity and correctness of digital signature. |
| Transaction ID | /transactionID | Obtain transaction ID tokens for Reliable Messaging. |

| | | |
|---|---|---|
| AIS Submit | /aisSubmit | Automated Import System. |
| AES Submit | /aesSubmit | Automated Export System. |
| NCTS Submit | /nctsSubmit | New Computerised Transit System. |
| EDE Submit | /edeSubmit | Excise Duty Entry. |
| EMCS Submit | /emcsSubmit | Excise Movement Control System.<br>**NOTE: Only web service URL is provided without actual backend EMCS functionality.** |
| Mailbox Collect | /mailboxCollect | Collect Customs and Excise responses. |
| Mailbox Acknowledge | /mailboxAcknowledge | Acknowledge the collection of Customs and Excise responses. |
| Export Release Verification | /export/releaseVerification | Submit ERV requests for Automated Export System. |

# 4 Connectivity Test Prerequisites

A developer or tester who wishes to engage in Connectivity Testing in PIT must first ensure that they have:

1. Notified Revenue on their intention to test through registering for access to the Revenue eCustoms PIT Support Service Desk
2. Received their ROS test digital certificate that will enable them to access and reset Revenue supplied test data from the PIT Test Data Management Service.

Further information on these tools, including access, will be provide on the PIT Homepage.

# 5 Test Scenarios for Synchronous Connectivity Testing

Synchronous Connectivity Testing establishes that the client can successfully interact with C&E endpoints, and consists of the following scenarios:

- SOAP POST handshake
- REST handshake with XML content
- REST handshake with JSON content

It is not required to complete all three scenarios – just the ones relevant to the specifics of implementer's solution.

For details on how to create and sign the requests, please refer to *"Customs & Excise SOAP Web Service Integration Guide"* and *"Customs & Excise REST Web Service Integration Guide"*.

## 5.1 SOAP POST Handshake

| | |
|---|---|
| **Test Identifier** | **SOAP POST handshake** |
| **Test Purpose** | Verify the Revenue responds with the expected synchronous response messages, following submission of the SOAP HandshakeRequest. |
| **URL live-level** | https://softwaretest.ros.ie/customs/webservice/v1/soap/handshake |
| **URL dev-level** | https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/handshake |
| **HTTP Method** | POST |
| **Content-type** | application/soap+xml |
| **Test Data Prerequisite** | Prepare valid HandshakeRequest message XML, signed with a valid ROS digital signature.<br>Sample HandshakeRequest payload:<br><ns2:HandshakeRequest<br>xmlns:ns2="http://www.ros.ie/schemas/customs/handshake"/> |
| **Test Steps** | 1. The Customs Agent/Trader prepares their Handshake Request XML message.<br>2. The Customs Agent/Trader submits the message to the SOAP Handshake webservice.<br>3. Revenue returns the relevant MessageAcknowledgement XML response message as listed below in Expected Results. |
| **Expected Result** | • Successful **Message Acknowledgement** – when the connectivity is established, and SOAP envelope signed correctly.<br>• ROS Error **Message Acknowledgement** – when the connectivity is established, but the SOAP envelope was not signed correctly or there is an issue with the webservice.<br>• **Any other response** – including empty, 404 HTTP status etc. – when the connectivity has not been established or the request is incorrect. |
| **Sample Result Payload** | Sample successful response payload:<br><ns2:MessageAcknowledgement<br>xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1"><br><ns2:Status><br><ns2:MessageStatus>SUCCESS</ns2:MessageStatus><br></ns2:Status><br></ns2:MessageAcknowledgement><br><br>Sample ROS error response payload:<br><ns2:MessageAcknowledgement<br>xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1"><br><ns2:ErrorReference><br><ns2:ErrorCode>ROS-100-10</ns2:ErrorCode><br></ns2:ErrorReference><br></ns2:MessageAcknowledgement><br><br>More information on error codes and common failure scenarios is available in *"Customs & Excise – Web Services Common Specification"* document. |

## 5.2 REST Handshake – XML Content

| | |
|---|---|
| **Test Identifier** | **REST handshake – XML content** |
| **Test Purpose** | Verify the Revenue responds with the expected synchronous response messages, following GET, POST or PUT calls to the REST handshake endpoint. |
| **Test URL live-level** | https://softwaretest.ros.ie/customs/webservice/v1/rest/handshake |
| **Test URL dev-level** | https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/handshake |
| **HTTP Methods** | GET, POST, PUT |
| **Content-type** | application/xml |
| **Test Data Prerequisite** | Prepare valid request message, signed with a valid ROS digital signature.<br>Sample request payloads:<br>• HTTP GET: "" (empty String)<br>• HTTP POST/PUT: <ns2:HandshakeRequest xmlns:ns2="http://www.ros.ie/schemas/customs/handshake"/> |
| **Test Steps** | 1. The Customs Agent/Trader prepares their message.<br>2. The Customs Agent/Trader submits the message to the REST Handshake webservice.<br>3. Revenue returns the relevant XML response as listed below in Expected Results. |
| **Expected Result** | • Successful **Message Acknowledgement** – when the connectivity is established, and the request signed correctly.<br>• ROS Error **Message Acknowledgement** – when the connectivity is established, but the request was not signed correctly or there is an issue with the webservice.<br>• **Any other response** – including empty, 4xx or 5xx HTTP status etc. – when the connectivity has not been established or the request is incorrect. |
| **Sample Result Payload** | Sample successful response:<br><ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1"><br><ns2:Status><br><ns2:MessageStatus>SUCCESS</ns2:MessageStatus><br></ns2:Status><br></ns2:MessageAcknowledgement><br><br>Sample ROS error response:<br><ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1"><br><ns2:ErrorReference><br><ns2:ErrorCode>ROS-100-10</ns2:ErrorCode><br></ns2:ErrorReference><br></ns2:MessageAcknowledgement><br><br>More information on error codes and common failure scenarios is available in *"Customs & Excise – Web Services Common Specification"* document. |

## 5.3 REST Handshake – JSON Content

| | |
|---|---|
| **Test Identifier** | **REST handshake – JSON content** |
| **Test Purpose** | Verify the Revenue responds with the expected synchronous response messages, following GET, POST or PUT calls to the REST handshake endpoint. |
| **Test URL live-level** | https://softwaretest.ros.ie/customs/webservice/v1/rest/handshake |
| **Test URL dev-level** | https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/handshake |
| **HTTP Methods** | GET, POST, PUT |
| **Content-type** | application/json |
| **Test Data Prerequisite** | Prepare valid request message, signed with a valid ROS digital signature.<br>Sample request payloads:<br>• HTTP GET: "" (empty String)<br>• HTTP POST/PUT: "Any string" |
| **Test Steps** | 1. The Customs Agent/Trader prepares their message.<br>2. The Customs Agent/Trader submits the message to the REST Handshake webservice.<br>3. Revenue returns the relevant JSON response as listed below in Expected Results. |
| **Expected Result** | • **Connection Status SUCCESS**– when the connectivity is established, and the request signed correctly.<br>• **Validation Errors** with ROS Error – when the connectivity is established, but the request was not signed correctly or there is an issue with the webservice.<br>• **Any other response** – including empty, 4xx or 5xx HTTP status etc. – when the connectivity has not been established or the request is incorrect. |
| **Sample Result Payload** | Sample successful response:<br><br>{"connectionStatus": "SUCCESS"}<br><br>Sample ROS error response:<br><br>{"validationErrors":[{"code":"ROS-300-10","description":"Issue with request's timestamp."}]}<br><br>More information on error codes and common failure scenarios is available in *"Customs & Excise – Web Services Common Specification"* document. |

# 6 Test Scenarios for Asynchronous End-to-End Interaction Testing

Testing of end-to-end interactions establishes that trader is receives and is able to collect an asynchronous response placed in trader's mailbox, e.g. IM428 and IM429 response messages to the original IM415 submission for AIS. These scenarios can be achieved via either SOAP XML or REST with XML body methods and are relevant for AIS, AES, NCTS-P5, EMCS, EDE systems.

They consist of three steps:

1. Submission of relevant Customs declaration.
2. Collection of asynchronous responses form trader's Mailbox.
3. Acknowledgment of the Collection of responses form trader's Mailbox.

## 6.1 Submit Customs Declaration

| | |
|---|---|
| **Test Identifier** | **Asynchronous End-to-End – Submit Customs Declaration – SOAP XML**<br>**Asynchronous End-to-End – Submit Customs Declaration – REST XML** |
| **Test Purpose** | Verify the Revenue responds with synchronous response messages providing Transaction ID, following submission of the standard Customs declaration relevant for the given C&E system. |
| **Test URLs live-level:** | SOAP:<br>• AIS: https://softwaretest.ros.ie/customs/webservice/v1/soap/aisSubmit<br>• AES: https://softwaretest.ros.ie/customs/webservice/v1/soap/aesSubmit<br>• NCTS-P5: https://softwaretest.ros.ie/customs/webservice/v1/soap/nctsSubmit<br>• EMCS: https://softwaretest.ros.ie/customs/webservice/v1/soap/emcsSubmit<br>• EDE: https://softwaretest.ros.ie/customs/webservice/v1/soap/edeSubmit<br>REST:<br>• AIS: https://softwaretest.ros.ie/customs/webservice/v1/rest/aisSubmit<br>• AES: https://softwaretest.ros.ie/customs/webservice/v1/rest/aesSubmit<br>• NCTS-P5: https://softwaretest.ros.ie/customs/webservice/v1/rest/nctsSubmit<br>• EMCS: https://softwaretest.ros.ie/customs/webservice/v1/rest/emcsSubmit<br>• EDE: https://softwaretest.ros.ie/customs/webservice/v1/rest/edeSubmit |
| **Test URLs development-level:** | SOAP:<br>• AIS: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/aisSubmit<br>• AES: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/aesSubmit<br>• NCTS-P5: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/nctsSubmit<br>• EMCS: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/emcsSubmit<br>• EDE: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/edeSubmit<br>REST:<br>• AIS: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/aisSubmit<br>• AES: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/aesSubmit<br>• NCTS-P5: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/nctsSubmit<br>• EMCS: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/emcsSubmit<br>EDE: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/edeSubmit |
| **HTTP Method** | POST |

| | | |
|---|---|---|
| **Content-type** | • SOAP: application/soap+xml | • REST: application/xml |
| **Test Data Prerequisite** | Prepare valid message for the system being tested, signed with a valid ROS digital signature. Sample message can be found on the PIT Homepage. | |
| **Test Steps** | 1. The Customs Agent/Trader prepares their customs declaration and signs with digital signature.<br>2. The Customs Agent/Trader submits the message to the relevant "submit" webservice.<br>3. Revenue responds synchronously with Message Acknowledgement containing Transaction ID token, used in the step "Collect Asynchronous Response from Mailbox" to identify asynchronous responses to this submission. | |
| **Expected Result** | **Message Acknowledgement** message containing Transaction ID token and status ACCEPTED. | |
| **Sample Result Payload** | <ns2:MessageAcknowledgement<br>xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1"><br><ns2:TransactionId>**d0b76be7-a2ae-454c-bce2-18d1fbffa371**</ns2:TransactionId><br><ns2:Status><br><ns2:MessageStatus>**ACCEPTED**</ns2:MessageStatus><br></ns2:Status><br></ns2:MessageAcknowledgement> | |

## 6.2 Collect Asynchronous Response from Mailbox

**Note on Mailboxes:**

Customs Mailbox is linked to the ROS digital certificate used to sign and submit the original Customs declarations. Therefore:

- Responses to declarations submitted using "Admin" certificate can be collected only by Mailbox Collect Request signed by this certificate, or by its renewed certificate.
- Responses to declarations submitted using "Sub-certificate" can be collected by Mailbox Collect Request signed only either by this sub-certificate (or by its renewed certificate) or its "Admin" certificate (or its renewed "Admin" certificate).

| | |
|---|---|
| **Test Identifier** | **Asynchronous End-to-End – Collect from Mailbox – SOAP XML**<br>**Asynchronous End-to-End – Collect from Mailbox – REST XML** |
| **Test Purpose** | Verify the Revenue responds with synchronous Mailbox Collect Response message containing asynchronous responses to previous submissions, following valid Mailbox Collect Request. |
| **Test URLs live-level** | SOAP: https://softwaretest.ros.ie/customs/webservice/v1/soap/mailboxCollect<br>REST: https:// softwaretest.ros.ie/customs/webservice/v1/rest/mailboxCollect |
| **Test URLs dev-level** | SOAP: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/mailboxCollect<br>REST: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/mailboxCollect |

| | | |
|---|---|---|
| **HTTP Method** | POST | |
| **Content-type** | • SOAP: application/soap+xml | • REST: application/xml |
| **Test Data Prerequisite** | Prepare valid Mailbox Collect Request message, signed with a valid ROS digital signature.<br>Sample Mailbox Collect Request:<br><ns2:MailboxCollectRequest<br>xmlns:ns2="http://www.ros.ie/schemas/customs/collectrequest/v1"/> | |
| **Test Steps** | 1. The Customs Agent/Trader prepares a valid Mailbox Collect Request message and signs with digital signature.<br>2. The Customs Agent/Trader submits the message to the Mailbox Collect webservice.<br>3. Revenue responds with Mailbox Collect Response. | |
| **Expected Result** | **Mailbox Collect Response**.<br>The response should contain expected asynchronous response message/messages to the originally submitted declaration, as triggered by the flow described in the relevant "message flows" document.<br>These asynchronous responses are identified by Transaction ID token. | |
| **Sample Result Payload** | <ns2:MailboxCollectResponse<br>xmlns:ns2="http://www.ros.ie/schemas/customs/collectresponse/v1"><br><ns2:MailboxItemList messagecount="5" moremessages="false"><br><ns2:MailboxItem><br><ns2:MailboxId>80663bd4-3526-417b-bc2e-b7852f940d4c</ns2:MailboxId><br><ns2:TransactionId>8f0ad209-6a53-49bd-852f-3448207a72e4</ns2:TransactionId><br><ns2:Message><br>// XML message contents<br></ns2:Message><br></ns2:MailboxItem><br><ns2:MailboxItem><br><ns2:MailboxId>**e5bcf0a4-1f7e-4f27-a68d-a808a74b322a**</ns2:MailboxId><br><ns2:TransactionId>**d0b76be7-a2ae-454c-bce2-18d1fbffa371**</ns2:TransactionId><br><ns2:Message><br>// XML message contents<br></ns2:Message><br></ns2:MailboxItem><br>. . . // Further Mailbox Items<br></ns2:MailboxItemList><br></ns2:MailboxCollectResponse><br><br>If the mailbox is empty, the response will look like so:<br><br><ns2:MailboxCollectResponse<br>xmlns:ns2="http://www.ros.ie/schemas/customs/collectresponse/v1"><br><ns2:MailboxItemList messagecount="0" moremessages="false"/><br></ns2:MailboxCollectResponse> | |

## 6.3 Acknowledge the Collection of Responses from Mailbox

| | |
|---|---|
| **Test Identifier** | **Asynchronous End-to-End – Mailbox Acknowledge – SOAP XML /**<br>**Asynchronous End-to-End – Mailbox Acknowledge – REST XML** |
| **Test Purpose** | Verify the Revenue responds with synchronous Mailbox Acknowledgement Response for a valid Mailbox Acknowledge Request, thus acknowledging the collection of specified asynchronous response messages and removing these responses form the Mailbox. |
| **Test URLs live-level** | SOAP: https:// softwaretest.ros.ie/customs/webservice/v1/soap/mailboxAcknowledge<br>REST: https:// softwaretest.ros.ie/customs/webservice/v1/rest/mailboxAcknowledge |
| **Test URLs dev-level** | SOAP: https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap/mailboxAcknowledge<br>REST: https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest/mailboxAcknowledge |
| **HTTP Method** | POST |
| **Content-type** | • SOAP: application/soap+xml<br>• REST: application/xml |
| **Test Data Prerequisite** | Prepare valid Mailbox Acknowledge Request message, signed with a valid ROS digital signature.<br>Sample Mailbox Acknowledge Request:<br><ns2:MailboxAcknowledgeRequest<br>xmlns:ns2="http://www.ros.ie/schemas/customs/acknowledgerequest/v1"><br><ns2:MailboxId>80663bd4-3526-417b-bc2e-b7852f940d4c</ns2:MailboxId><br><ns2:MailboxId>**e5bcf0a4-1f7e-4f27-a68d-a808a74b322a**</ns2:MailboxId><br>. . . // Further Mailbox IDs<br></ns2:MailboxAcknowledgeRequest><br>Where **MailboxId** specifies the ID under which the collected response resides, as returned in Mailbox Collect Response (see "Collect Asynchronous Response from Mailbox" above) |
| **Test Steps** | 1. The Customs Agent/Trader prepares a valid Mailbox Acknowledge Request message and signs with digital signature.<br>2. The Customs Agent/Trader submits the message to the Mailbox Acknowledge webservice.<br>3. Revenue responds with Mailbox Acknowledge Response. |
| **Expected Result** | **Mailbox Collect Response.**<br>It will specify which Mailbox Ids were successfully acknowledged and removed from Mailbox (Status SUCCESS). Non-existing Mailbox Ids (e.g. ones that had been acknowledged previously) will have status NOT FOUND. |
| **Sample Result Payload** | <ns2:MailboxAcknowledgeResponse<br>xmlns:ns2="http://www.ros.ie/schemas/customs/acknowledgeresponse/v1"><br><ns2:MailboxAcknowledgementList><br><ns2:MailboxAcknowledgement><br><ns2:MailboxId>80663bd4-3526-417b-bc2e-b7852f940d4c</ns2:MailboxId><br><ns2:AcknowledgementStatus>**NOT_FOUND**</ns2:AcknowledgementStatus><br></ns2:MailboxAcknowledgement><br><ns2:MailboxAcknowledgement><br><ns2:MailboxId>**e5bcf0a4-1f7e-4f27-a68d-a808a74b322a**</ns2:MailboxId><br><ns2:AcknowledgementStatus>**SUCCESS**</ns2:AcknowledgementStatus><br></ns2:MailboxAcknowledgement><br>. . . // Further Mailbox Acknowledgements<br></ns2:MailboxAcknowledgementList><br></ns2:MailboxAcknowledgeResponse> |