Revenue
Cáin agus Custaim na hÉireann
Irish Tax and Customs

Customs & Excise

Web Services
Common Specification

Contents

Version Control		
Version	**Date**	**Change**
1.0		Initial document published

Document References

Reference
1. *Customs & Excise SOAP Web Service Integration Guide*
2. *Customs & Excise REST Web Service Integration Guide*
3. *Customs & Excise Web Services Specifications for Payer and Importer Reports*
4. *Customs & Excise Web Services Specifications for C&E Enquiries*

Abbreviations Used in This Document

Abbreviation	Description
AES	Automated Export System
AIS	Automated Import System
C&E	Customs and Excise
CCI	Centralised Clearance for Import
EDE	Excise Duty Entry
EMCS	Excise Movement Control System
EORI	Economic Operators Registration and Identification
ERV	Export Release Verification
LOLO	Lift-On Lift-Off
NCTS	New Computerised Transit System
PIT	Public Interface Testing
RORO	Roll-On Roll-Off system
ROS	Revenue Online Service
UCC	Union Customs Code

Audience

This document is for any Trader/Economic Operator and/or software provider who has chosen to build or update their products to consume web services provided by Revenue Customs and Excise division to interact with various Customs systems.

1 Introduction

This document details the Customs & Excise ROS web services specification including the AIS (Import; including CCI), AES (Export), NCTS (Transit) and legacy web services migrated to the current SOAP specification (EMCS, EDE).

This document supersedes the previous specification document *"Web Services Specifications (incl. AES Operational January 2023 and AIS)"*, version 0.4, published on 28/01/2022.

The Revenue Customs web services are exposed over SOAP and/or REST, depending on the specific web service. For further details please refer to the Document References section.

ROS web services are built on industry standard technologies. They are available over the Internet and ensure the same level of privacy and security as the ROS web site.

This document covers the following Customs & Excise web services:

Service	Type	Endpoint URL	Additional Information
Handshake	REST SOAP	/handshake	General handshake to verify connectivity and correctness of digital signature.
Transaction ID	REST SOAP	/transactionID	Obtain transaction ID tokens for Reliable Messaging.
AIS Submit	REST SOAP	/aisSubmit	Automated Import System
AES Submit	REST SOAP	/aesSubmit	Automated Export System
NCTS Submit	REST SOAP	/nctsSubmit	New Computerised Transit System
EDE Submit	REST SOAP	/edeSubmit	Excise Duty Entry.
EMCS Submit	REST SOAP	/emcsSubmit	Excise Movement Control System.
Mailbox Collect	REST SOAP	/mailboxCollect	Collect Customs and Excise response.
Mailbox Acknowledge	REST SOAP	/mailboxAcknowledge	Acknowledge the collection of Customs and Excise responses.
Export Release Verification	REST SOAP	/export/releaseVerification	Submit ERV requests.

It does not cover C&E Payer/Importer Reports and Enquiries web services, which are described in their own documents, as specified in the Document References section.

2 Calling the Services

To call a web service one must send to a valid URL a request containing valid payload signed with ROS certificate as described below.

The full web service URL consists of environment-specific URL and the service-specific endpoint, e.g.:

`https://www.ros.ie/customs/webservice/v1/rest/aisSubmit`

Environment URLs:

Live:	https://www.ros.ie/customs/webservice/v1/rest https://www.ros.ie/customs/webservice/v1/soap
Public Interface Test (PIT) Live-level:	https://softwaretest.ros.ie/customs/webservice/v1/rest https://softwaretest.ros.ie/customs/webservice/v1/soap
Public Interface Test (PIT) Development-level:	https://softwaretestnextversion.ros.ie/customs/webservice/v1/rest https://softwaretestnextversion.ros.ie/customs/webservice/v1/soap

2.1 Digital Signatures

Any ROS web service request that either returns confidential information or accepts submission of information must be digitally signed. This must be done using a digital certificate that has been previously assigned to Trader/Economic Operator and retrieved from ROS.

The digital signature must be applied to the message in accordance with the modified HTTP Signatures specification as described in *"Customs & Excise REST Web Service Integration Guide"* and *"Customs & Excise SOAP Web Service Integration Guide"* available on Revenue.ie website.

The digital signature ensures the integrity of the document. By signing the document, we can ensure that no malicious intruder has altered the document in any way. It can also be used for non-repudiation purposes.

If a valid digital signature is not attached, a HTTP status code of 401 (Unauthorised) will be returned. The message body will provide more information on the details of the problem.

For REST GET requests the message to be signed is an empty string.

2.2 HTTP Content-type Header Values

The table below specifies allowed values for HTTP content-type header:

Content-type value	Purpose
application/soap+xml	All SOAP requests
application/soap+xml;charset=utf-8	All SOAP requests
application/xml	REST POST/PUT request with XML payload
application/json	REST GET/ POST/PUT request with JSON or empty string payload
application/json;charset=utf-8	REST GET/ POST/PUT request with JSON or empty string payload

2.3 Synchronous Responses

2.3.1 Overview

Each web service returns a synchronous response message to the client. The type of this synchronous response depends on:

- Functional nature of the web service (e.g. AIS Submit vs Mailbox Collect)
- Success vs error response
- Type of errors (Authentication/authorisation error vs functional/validation error)

This is outlined in the table below:

Service	Success Response	Error Response	Notes
Handshake; SOAP or REST with XML payload	Message Acknowledgement with Status SUCCESS	Message Acknowledgement with ROS Error	
Handshake; REST with JSON payload	*{"connectionStatus": "SUCCESS"}*	Validation Errors object with ROS Error	
AIS, AES, NCTS, EDE, EMCS Submit	Message Acknowledgement containing Transaction ID	Message Acknowledgement with ROS Error	ROS Error is returned for authentication, authorisation and internal service errors. Functional errors in submitted messages are returned as responses in Customs Mailbox
Mailbox Collect	Mailbox Collect Response	Message Acknowledgement with ROS Error	ROS Error is returned for authentication, authorisation and internal service errors. There are no functional errors.
Mailbox Acknowledge	Mailbox Acknowledge Response	Message Acknowledgement with ROS Error	ROS Error is returned for authentication, authorisation and internal service errors. There are no functional errors.
Transaction ID	Transaction ID Response	Message Acknowledgement with ROS Error	Returned for authentication, authorisation and internal service errors.

Transaction ID	Transaction ID Response	Transaction ID Response with functional error	Returned for functional errors.
Export Release Verification	Export Release Verification Response	Message Acknowledgement with ROS Error	Returned for authentication, authorisation and internal service errors.
Export Release Verification	Export Release Verification Response	Export Release Verification Response	Returned for functional errors.

2.3.2 Common Failure Scenarios

These are the most common failure scenarios that may arise when sending messages between a Trader/Economic Operator and ROS. In the event of a connection failure when the Trader/Economic Operator attempts to communicate with ROS or ROS attempts to communicate with the Trader/Economic Operator, then connection will time out. In case where the connection is successful, but other condition in encountered where the message cannot be processed, ROS returns an appropriate error code.

ROS will respond with an error message if any of the following occur:

- The message is not well formed.
- The request is for an invalid web service.
- The digital signature cannot be validated.
- The trader is not authorised to file messages to the specific web service.
- The type of SOAP message being submitted is not recognised.

For further details in relation to the ROS Error Codes please refer to the List of ROS Error Codes section below.

2.4 Asynchronous Responses

ROS Customs submissions (AIS, AES, NCTS, EMCS, EDE), apart from the synchronous "message acknowledgement" responses, also result in an asynchronous functional response messages or error messages generated by the respective Customs system (e.g. IM428 response to submitted earlier IM415 Import Declaration). These will be placed in the Mailbox for asynchronous collection; see Asynchronous Response Delivery Process section below.

3 List of Common ROS Error Codes

Common ROS Error codes possible in responses are presented in the table below.

Error Code	Applicability	Description
ROS-300-02	REST / SOAP	Issue with requests media type.
ROS-300-03	SOAP	Issue with parsing SOAP message. The message might be malformed.
ROS-300-10	REST / SOAP	Issue with the request's timestamp.
ROS-300-20	REST / SOAP	Issue with request's digital signature.
ROS-300-30	REST / SOAP	Issue with request's digest.
ROS-300-40	REST / SOAP	Certificate holder does not have permissions to submit this request.
ROS-100-00	REST / SOAP	Unrecognised digital certificate used.
ROS-100-10	REST / SOAP	Digital certificate used to sign the request is expired.
ROS-100-20	REST / SOAP	Digital certificate used to sign the request is revoked.
ROS-100-30	REST / SOAP	Digital certificate used to sign the request is invalid.
FRQ-100-10	REST / SOAP	Request submitted too soon after the previous one.
REL-100-10	REST / SOAP	Transaction ID request was invalid.
ROS-300-00	REST / SOAP	Unexpected error in processing the request. Please try again later.

4 Message Delivery Process

This section details the exchange of messages between the Trader/Economic Operator software and ROS. There are two default methods of message exchange, without reliable messaging and with reliable messaging. Both are outlined below.

4.1 Default Message Delivery Process (without Reliable Messaging)

After the Trader/Economic Operator creates and sends the digitally signed message to ROS, ROS verifies that the message is well formed and that the digital signature is valid. Once ROS verifies that the type of message is acceptable to the web service and the holder of the digital certificate used to sign the message is authorised to file messages of the given type, ROS accepts the message for processing and replies synchronously with an acknowledgement containing the transaction ID.

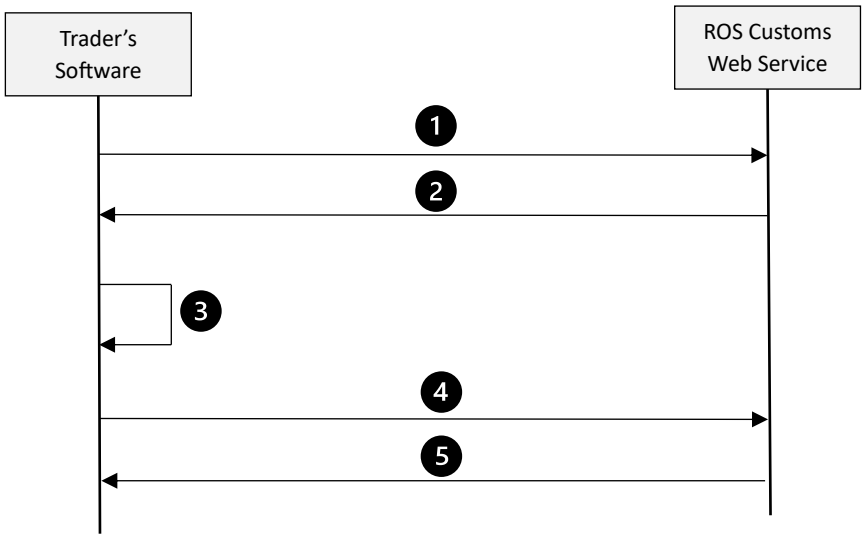
4.2 Message Delivery Process with Reliable Messaging

Reliable messaging is an **optional piece of functionality** that applies to the "submit" services (AIS, AES, NCTS, EMCS, EDE). In the context of the Customs & Excise web services, reliable messaging ensures that a message submitted was received by Revenue and that a message is processed only once by Revenue.

When calling one of the submit web services without reliable messaging, the Trader/Economic Operator software submits the request and receives an acknowledgement from ROS. However, if this acknowledgement is not properly received (for example, due to a lost network connection or a timeout) then the Trader/Economic Operator software cannot positively determine whether the message was received by ROS or not, i.e. did the connection fail before the message reached ROS, or after it reached ROS and before the acknowledgement was returned. There are two choices for the Trader/Economic Operator at this point: resend the message and risk it being processed a second time, or do not resend the message and risk it not being received at all by ROS.

The reliable messaging protocol serves to address this issue.

The usage is shown on the diagram below.



1) The Trader/Economic Operator requests from the "Transaction ID" web service up to 100 Transaction Ids in advance of calling any of the submission web services.
2) "Transaction ID" web service generates unique transaction ID tokens and returns them to the Trader/Economic Operator.
3) The Trader/Economic Operator software assigns each of these transaction IDs to a Customs submission request (AIS, AES, NCTS, EMCS, EDE)
4) The Trader/Economic Operator software sends the assigned Transaction ID with the request when calling the submission web service. This is done by either:
   - For SOAP requests: including "Transaction" tag inside SOAP Envelope Header element containing the Transaction ID value, e.g.:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
  ... other SOAP header elements omitted ...
  <rms:Transaction
  xmlns:rms="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">
    e6c3ce61-6836-4949-af66-90d3daf13914
  </rms:Transaction>
</env:Header>
```

   - For REST requests: including "Transaction" Http Header containing the Transaction ID value, e.g.:

```
"Transaction":"e6c3ce61-6836-4949-af66-90d3daf13914"
```

5) The submission web service ensures that the Transaction ID is valid (it was issued by ROS and has not yet expired) and ensure that each ID is processed only once and returns message acknowledgement.

Thus, for the scenario described above where an acknowledgement is not received by the Trader/Economic Operator Software, it can safely resend the same Customs submission request with the same Transaction ID as ROS will guarantee that duplicate submissions with the same Transaction ID are not processed twice.

For the second (and subsequent) submissions with the same given Transaction ID, message acknowledgement response is returned synchronously to acknowledge successful or failed delivery of the message to Irish Customs & Excise for processing.

- A successful "MessageAcknowledgement" response, apart from the standard "MessageStatus" tag, will contain additional "TransactionIdStatus" tag:

```
<ns2:MessageAcknowledgement
xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">
  <ns2:TransactionId>e6c3ce61-6836-4949-af66-90d3daf13914</ns2:TransactionId>
  <ns2:Status>
    <ns2:MessageStatus>ACCEPTED</ns2:MessageStatus>
    <ns2:TransactionIdStatus>ALREADY ACCEPTED</ns2:TransactionIdStatus>
  </ns2:Status>
</ns2:MessageAcknowledgement>
```

There are a number of different scenarios for the acknowledgment response document to take when the trader uses Reliable Messaging. They are listed in the table below.

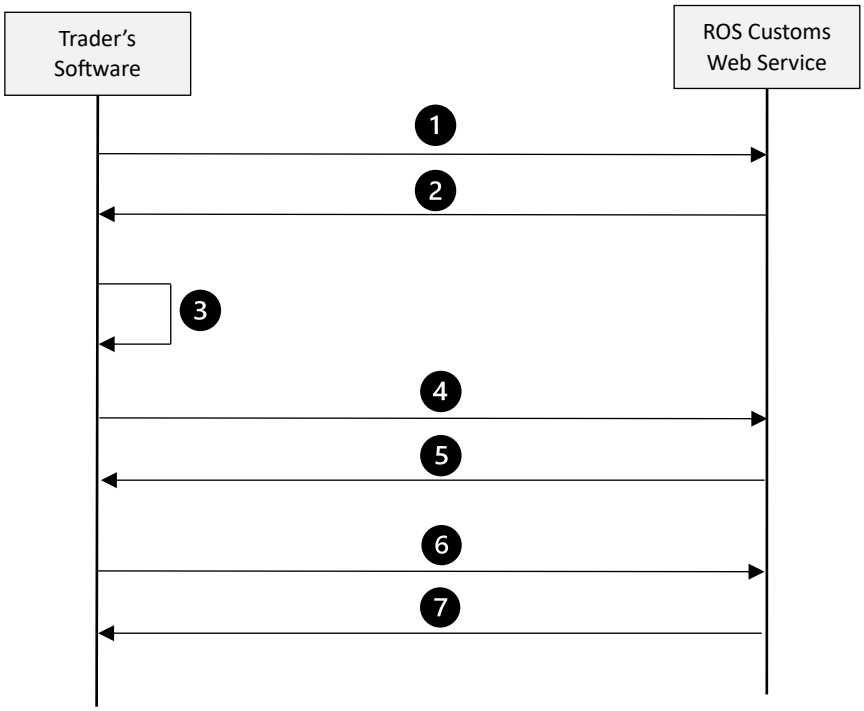
Message Status	Transaction ID Status	Description
ACCEPTED	ACCEPTED	The message has been accepted and the transaction ID is valid. The request is sent for processing.
ACCEPTED	ALREADY_ACCEPTED	The message has been accepted but the transaction ID has previously been used by another Customs & Excise submission. The request is not sent for processing.
REJECTED	INVALID	The message has been rejected because the transaction ID is invalid. This can happen if the Transaction ID send with the request was not retrieved from Revenue in the first place.
REJECTED	EXPIRED	The Transaction ID used is recognised, has not been used previously, but the message has been rejected because this transaction ID has expired, i.e. it is more than 7 days old. A new Transaction ID should be retrieved and used.
REJECTED	ERROR	There was an error with the submission of the message unrelated to the Transaction ID; see the ErrorReference element for more details.

- A negative "MessageAcknowledgement" response will contain an error code indicating the reason for the failure, as per standard message acknowledgement with error.

5 Asynchronous Response Delivery Process

5.1 Introduction

The process described below pertains to the collection of asynchronous responses from Customs AIS, AES, NCTS, EMCS and/or EDE systems, e.g. collection of IM428 response to submitted earlier IM415 Import Declaration.



1) The Trader/Economic Operator sends a submission to one of the ROS Customs "submit" services (AIS, AES, NCTS, EMCS, EDE).
2) "Submit" service responds synchronously with transaction ID.
3) The Trader/Economic Operator software tracks the transaction ID for the submission.
4) The Trader/Economic Operator sends the digitally signed "Mailbox Collect" message to ROS "Mailbox Collect" service.
5) "Mailbox Collect" service determines the ID of the Trader/Economic Operator from the digital certificate used to sign the request and ensures that the Trader/Economic Operator hasn't made a collect request within the last 2 seconds. "Mailbox Collect" service then responds synchronously with a list of maximum 50 previously uncollected and unacknowledged mailbox items to the Trader/Economic Operator. Each item is identified by unique "mailbox ID" as well as "transaction ID" which links the response to the original submission.
   The "Mailbox Collect" response also indicates whether there are more items in the mailbox awaiting collection.

6) The Trader/Economic Operator sends a Mailbox Acknowledgement message containing the list of mailbox IDs to ROS "Mailbox Acknowledge" service to confirm collection of the specified items.
7) "Mailbox Acknowledge" service removes the items from the mailbox and synchronously responds with a list of the IDs and statuses of all acknowledged (removed), thus confirming that the specified items were acknowledged correctly.

5.2 Note

The Mailbox service works based on the digital certificates used to sign the original Customs submission messages as well as the Mailbox Collect/Acknowledge requests. There are two types of certificates: Admin and sub-cert. The behaviour of Mailbox Collect depends on which one is used to sign the Mailbox requests:

- Signing the Mailbox Collect request with an **Admin certificate will return from the mailbox all items that belong to the Trader/Economic Operator who is the holder of this Admin certificate.** I.e., if the Trader/Economic Operator submitted requests to different Customs systems (e.g. AIS IM415 and AES IE515/CC515C) the Mailbox Collect response will contain responses to both AIS and AES submissions.

- Signing the Mailbox Collect request with a **sub cert will return from the mailbox only the items that are responses to Customs submissions that were signed with this sub-cert.** I.e., if the Trader/Economic Operator uses sub-cert A to sign submissions to AIS and sub-cert B to sign submissions to AES, Mailbox Collect calls signed with sub-cert A will return only responses from AIS, while Mailbox Collect calls signed with sub-cert B will return only responses from AES.

Therefore, Revenue recommendation is to use dedicated, separate sub-certs for each Customs system a Trader/Economic Operator interacts with.

5.3 Failure Scenarios

ROS detects that there has been a collect request from the Trader/Economic Operator within the last 2 seconds:

- ROS responds with error code FRQ-100-10 (Request submitted too soon after the previous one)

ROS is unable to complete the mailbox operation:

- ROS responds with a relevant error code. See Common Failure Scenarios.

6 Web Services

The following sections describe the operation of each web service.

Below is a list of web services that can be used to indicate that the relevant message should be submitted to Irish Customs & Excise. The payloads of these web service messages (except for the Handshake) are always in XML format according to the schemas relevant to the appropriate service and/or Customs system. Please refer to the section on the Revenue website pertaining to the given system for details on the schemas.

Following a successful request, the message will be processed, and ROS Customs web service will respond with an appropriate synchronous response. If relevant, appropriate asynchronous response message from the relevant Customs system will be placed in the customer's mailbox, as described in Asynchronous Response Delivery Process section above.

6.1 Handshake

6.1.1 SOAP Handshake

Purpose	Validate connectivity and digital signature for SOAP request.
Content-type	`application/soap+xml`
Endpoint	`/handshake`
Request payload	`<ns2:HandshakeRequest xmlns:ns2="http://www.ros.ie/schemas/customs/handshake"/>`

Synchronous successful response	`<ns2:MessageAcknowledgement`
	`xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>SUCCESS</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement`
	`xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.1.2 REST Handshake with XML Content

Purpose	Validate connectivity and digital signature for REST calls with XML content.
Method	GET, POST, PUT
Content-type	`application/xml`
Endpoint	`/handshake`
Request payload	**HTTP GET:** `""` (Empty string) **HTTP POST / PUT:** `<ns2:HandshakeRequest xmlns:ns2="http://www.ros.ie/schemas/customs/handshake"/>`

Synchronous successful response	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`<ns2:Status>`
	`<ns2:MessageStatus>SUCCESS</ns2:MessageStatus>`
	`</ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`<ns2:ErrorReference>`
	`<ns2:ErrorCode>`***ERROR-CODE-VALUE***`</ns2:ErrorCode>`
	`</ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.1.3 REST Handshake with JSON Content

Purpose	Validate connectivity and digital signature for REST calls with JSON content.
Http Method	GET, POST, PUT
Content-type	`application/json`
Endpoint	`/handshake`
Request payload	**HTTP GET:** `""` (Empty string) **HTTP POST / PUT:** `"Anything"` (Any string)
Synchronous successful response	`{"connectionStatus": "SUCCESS"}`

Synchronous error response format	`{"validationErrors":[{`
	`"code":"string",`
	`"description":"string"`
	`}]}`

6.2 Transaction ID

Purpose	Obtain specified number of Transaction ID Tokens for use with Reliable Messaging.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/transactionID`

Request payload	`<ns2:TransactionIDRequest xmlns:ns2="http://www.ros.ie/schemas/customs/transactionidrequest/v1">`
	`  <ns2:Transactions>`
	`  <ns2:NumberOfTxIds>`***N***`</ns2:NumberOfTxIds>`
	`  </ns2:Transactions>`
	`</ns2:TransactionIDRequest>`
	
	Where ***N*** is an integer between 1 and 100 inclusive.

Synchronous successful response format	`<ns2:TransactionIDResponse xmlns:ns2="http://www.ros.ie/schemas/customs/transactionidresponse/v1">`
	`  <ns2:Transactions>`
	`  <ns2:TransactionId>`***token-value-1***`</ns2:TransactionId>`
	`  <ns2:TransactionId>`***token-value-2***`</ns2:TransactionId>`
	`  . . .`
	`  <ns2:TransactionId>`***token-value-n***`</ns2:TransactionId>`
	`  </ns2:Transactions>`
	`</ns2:TransactionIDResponse>`

Synchronous error response format	Transaction ID response in case of functional error:
	
	`<ns3:TransactionIDResponse xmlns:ns3="http://www.ros.ie/schemas/customs/transactionidresponse/v1">`
	`  <ns3:ErrorReference>`
	`  <ns3:ErrorCode>`***ERROR-CODE-VALUE***`</ns3:ErrorCode>`
	`  </ns3:ErrorReference>`
	`</ns3:TransactionIDResponse>`
	
	Message Acknowledgement with error in case of other errors, e.g. authorisation:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***ERROR-CODE-VALUE***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.3 AIS Submit

Purpose	Submit declarations to Automated Import System.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/aisSubmit`
Request payload	Message of types listed below as per AIS schema: IM415, IM414, IM413, IM432, IM483, ETD13, ETD14, ETD15, RF415, RD415, TS313, TS314, TS315, TS332

Synchronous successful response format	Without Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`
	
	With Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  <ns2:TransactionIdStatus>`***`STATUS_VALUE`***`</ns2:TransactionIdStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.4 AES Submit

Purpose	Submit declarations to Automated Export System.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/aesSubmit`
Request payload	Message of types listed below as per AES schema: CC507C (IE507) , CC511C (IE511) , CC513C (IE513) , CC514C (IE514) , CC515C (IE515) , CC570C (IE570) , CC573C (IE573) , CC583C (IE583) , CC590C (IE590) , CC613C (IE613) , CC614C (IE614) , CC615C (IE615) , EX583 (EX583) , EX513 (EX513)

Synchronous successful response format	Without Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`
	
	With Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  <ns2:TransactionIdStatus>`***`STATUS_VALUE`***`</ns2:TransactionIdStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.5 NCTS Submit

Purpose	Submit declarations to New Computerised Transit System.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/nctsSubmit`
Request payload	Message of types listed below as per NCTS schema: CC007C (IE007) , CC013C (IE013) , CC014C (IE014) , CC015C (IE015) , CC026C (IE026) , CC034C (IE034) , CC044C (IE044) , CC054C (IE054) , CC141C (IE141) , CC170C (IE170) , CC224C (IE224) , TR083C (TR083)

Synchronous successful response format	Without Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`
	
	With Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***`transaction-id-value`***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***`STATUS_VALUE`***`</ns2:MessageStatus>`
	`  <ns2:TransactionIdStatus>`***`STATUS_VALUE`***`</ns2:TransactionIdStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.6 EDE Submit

Purpose	Submit Excise Duty Entries.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/edeSubmit`
Request payload	Excise Duty Entry messages as "ImportCustomsDeclaration" defined by "http://www.ros.ie/schemas/customs/sadimport/v1" schema.

Synchronous successful response format	Without Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***transaction-id-value***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***STATUS_VALUE***`</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`
	
	With Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>`***transaction-id-value***`</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>`***STATUS_VALUE***`</ns2:MessageStatus>`
	`  <ns2:TransactionIdStatus>`***STATUS_VALUE***`</ns2:TransactionIdStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***ERROR-CODE-VALUE***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.7 EMCS Submit

Purpose	Submit declarations to Excise Movement Control System.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/nctsSubmit`
Request payload	Message of types listed below as per EMCS schema: • IE810 (CancellationOfAnEAAD), • IE813 (ChangeOfDestination), • IE815 (SubmittedDraftOfEAAD), • IE818 (AcceptedOrRejectReportOfReceipt), • IE819 (AlertOrRejectionOfAnEAAD), • IE825 (SubmittedDraftOfSplittingOperation), • IE837 (ExplanationOnDelayForDelivery), • IE871 (ExplanationOnReasonForShortage)

Synchronous successful response format	Without Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>transaction-id-value</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>STATUS_VALUE</ns2:MessageStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`
	
	With Reliable Messaging:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:TransactionId>transaction-id-value</ns2:TransactionId>`
	`  <ns2:Status>`
	`  <ns2:MessageStatus>STATUS_VALUE</ns2:MessageStatus>`
	`  <ns2:TransactionIdStatus>STATUS_VALUE</ns2:TransactionIdStatus>`
	`  </ns2:Status>`
	`</ns2:MessageAcknowledgement>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>ERROR-CODE-VALUE</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.8 Mailbox Collect

Purpose	Collect asynchronous responses from Customs mailbox.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/ mailboxCollect`
Request payload	`<ns2:MailboxCollectRequest xmlns:ns2="http://www.ros.ie/schemas/customs/collectrequest/v1"/>`

Synchronous successful response format	With messages retrieved from Customs mailbox:
	
	`<ns2:MailboxCollectResponse xmlns:ns2="http://www.ros.ie/schemas/customs/collectresponse/v1">`
	`  <ns2:MailboxItemList messagecount="N" moremessages="true/false">`
	`  <ns2:MailboxItem>`
	`  <ns2:MailboxId>mailbox-id-value-1</ns2:MailboxId>`
	`  <ns2:TransactionId>transaction-id-1</ns2:TransactionId>`
	`  <ns2:Message>`
	`  // XML message contents`
	`  </ns2:Message>`
	`  </ns2:MailboxItem>`
	`  <ns2:MailboxItem>`
	`  <ns2:MailboxId>mailbox-id-value-2</ns2:MailboxId>`
	`  <ns2:TransactionId>transaction-id-2</ns2:TransactionId>`
	`  <ns2:Message>`
	`  // XML message contents`
	`  </ns2:Message>`
	`  </ns2:MailboxItem>`
	`  . . . // Further Mailbox Items`
	`  </ns2:MailboxItemList>`
	`</ns2:MailboxCollectResponse>`
	
	With Mailbox empty:
	
	`<ns2:MailboxCollectResponse xmlns:ns2="http://www.ros.ie/schemas/customs/collectresponse/v1">`
	`  <ns2:MailboxItemList messagecount="0" moremessages="false"/>`
	`</ns2:MailboxCollectResponse>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>ERROR-CODE-VALUE</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.9 Mailbox Acknowledge

Purpose	Acknowledge collection of asynchronous responses from Customs mailbox to enable collection of further messages from mailbox.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/ mailboxAcknowledge`

Request payload	`<ns2:MailboxAcknowledgeRequest xmlns:ns2="http://www.ros.ie/schemas/customs/acknowledgerequest/v1">`
	`  <ns2:MailboxId>`***`mailbox-id-value-1`***`</ns2:MailboxId>`
	`  <ns2:MailboxId>`***`mailbox-id-value-2`***`</ns2:MailboxId>`
	`  . . . // Further Mailbox IDs`
	`  <ns2:MailboxId>`***`mailbox-id-value-n`***`</ns2:MailboxId>`
	`</ns2:MailboxAcknowledgeRequest>`

Synchronous successful response format	`<ns2:MailboxAcknowledgeResponse xmlns:ns2="http://www.ros.ie/schemas/customs/acknowledgeresponse/v1">`
	`  <ns2:MailboxAcknowledgementList>`
	`  <ns2:MailboxAcknowledgement>`
	`  <ns2:MailboxId>`***`mailbox-id-value-1`***`</ns2:MailboxId>`
	`  <ns2:AcknowledgementStatus>`***`STATUS_VAL`***`</ns2:AcknowledgementStatus>`
	`  </ns2:MailboxAcknowledgement>`
	`  <ns2:MailboxAcknowledgement>`
	`  <ns2:MailboxId>`***`mailbox-id-value-2`***`</ns2:MailboxId>`
	`  <ns2:AcknowledgementStatus>`***`STATUS_VAL`***`</ns2:AcknowledgementStatus>`
	`  </ns2:MailboxAcknowledgement>`
	`  . . . // Further Mailbox Acknowledgements`
	`  <ns2:MailboxAcknowledgement>`
	`  <ns2:MailboxId>`***`mailbox-id-value-n`*** `</ns2:MailboxId>`
	`  <ns2:AcknowledgementStatus>`***`STATUS_VAL`***`</ns2:AcknowledgementStatus>`
	`  </ns2:MailboxAcknowledgement>`
	`  </ns2:MailboxAcknowledgementList>`
	`</ns2:MailboxAcknowledgeResponse>`

Synchronous error response format	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`

6.10 Export Release Verification

Purpose	Simplified 'Arrival at exit' (IE507) and 'Confirmation of exit' (IE590) submission for Air and LOLO based Carriers and their Agents.
Type	SOAP / REST
Content-type	SOAP calls: `application/soap+xml` REST calls: `application/xml`
Endpoint	`/export/releaseVerification`

Request payload	Export Release Verification Request message as per ERV schema.
	
	`<erv:ExportReleaseVerificationRequest xmlns:erv="http://www.ros.ie/schemas/customs/exportReleaseVerificationRequest/v1">`
	`  <erv:RequestType>`***`TYPE_VALUE`***`</erv:RequestType>`
	`  <erv:MRN>`***`MRN_VALUE`***`</erv:MRN>`
	`  <erv:Agent>`***`AGENT_EORI`***`</erv:Agent>`
	`  <erv:Carrier>`***`CARRIER_EORI`***`</erv:Carrier>`
	`</erv:ExportReleaseVerificationRequest>`

Synchronous successful response format	Export Release Verification Response message as per ERV schema.
	Response to release request:
	
	`<erv:ExportReleaseVerificationResponse xmlns:erv="http://www.ros.ie/schemas/customs/exportReleaseVerificationResponse">`
	`  <MRN>`***`MRN_VALUE`***`</MRN>`
	`  <Status>`***`STATUS_VALUE`***`</Status>`
	`  <DepartureConfirmationRequired>`***`true/false`***`</DepartureConfirmationRequired>`
	`  <TransactionId>`***`transaction-id-value`***`</TransactionId>`
	`</erv:ExportReleaseVerificationResponse>`
	
	Response to departure confirmation:
	
	`<erv:ExportReleaseVerificationResponse xmlns:erv="http://www.ros.ie/schemas/customs/exportReleaseVerificationResponse">`
	`  <MRN>`***`MRN_VALUE`***`</MRN>`
	`  <Status>`***`STATUS_VALUE`***`</Status>`
	`  <TransactionId>`***`transaction-id-value`***`</TransactionId>`
	`</erv:ExportReleaseVerificationResponse>`

Synchronous error response format	Export Release Verification Response in case of functional error:
	
	`<erv:ExportReleaseVerificationResponse xmlns:erv="http://www.ros.ie/schemas/customs/exportReleaseVerificationResponse">`
	`  <Status>`**`REJECTED`**`</Status>`
	`  <Error>`
	`  <Code>`***`ERROR-CODE-VALUE`***`</Code>`
	`  <Description>`***`ERROR DESCRIPTION`***`</Description>`
	`  </Error>`
	`</erv:ExportReleaseVerificationResponse>`
	
	Message Acknowledgement with error in case of other errors, e.g., authorisation:
	
	`<ns2:MessageAcknowledgement xmlns:ns2="http://www.ros.ie/schemas/customs/messageacknowledgement/v1">`
	`  <ns2:ErrorReference>`
	`  <ns2:ErrorCode>`***`ERROR-CODE-VALUE`***`</ns2:ErrorCode>`
	`  </ns2:ErrorReference>`
	`</ns2:MessageAcknowledgement>`