DMS System Guide

Functional guide to DMS

UDVIKLINGS OG FORENKLINGS STYRELSEN

Table of contents

Introduction

1

This guide describes the functionality of the new Declaration Management System – DMS. The target group for this system guide is developers responsible for developing system-to-system (S2S) integration from own customs clearance system to DMS System-to-System.

The aim of this document is to provide an understanding of the technical setup around a system-to-system integration. While the DMS Connectivity Guide (found on Danish Customs and Tax Administration's GitHub) explains how to establish connection to the AS4-gateway, this system guide explains the message flows and the functions that can be carried out with DMS. In specific, it explains the notifications in detail.

This document will be enhanced continuously. So far, the document covers functionality of DMS in general around DMS Import, DMS Export and DMS Transit. Specifically, on DMS Import only H7 and I2 declaration types are covered. For DMS Export specifically B1, B2, B4, C1 and C2 declaration types are covered.

Technical overview

2

2.1 Relation between DMS System-to-System and DMS Online

DMS can be accessed either via an S2S solution called DMS System-to-System, where declarations are submitted through the AS4-gateway, or via the systems online UI called DMS Online.

In principle, all system functions can be managed through both access points, UI or S2S. If using DMS System-to-System, the recommendation is to use the UI only to look up information to avoid the risk of mismatch of data between own backend and DMS. E.g., if a declaration is lodged via S2S integration but amended via the UI, your backend will not know which data was changed via the UI.

2.2 System overview

DMS has two types of actions: One for submitting declarations and additional messages and one for requesting notifications, which are statuses of a declaration.

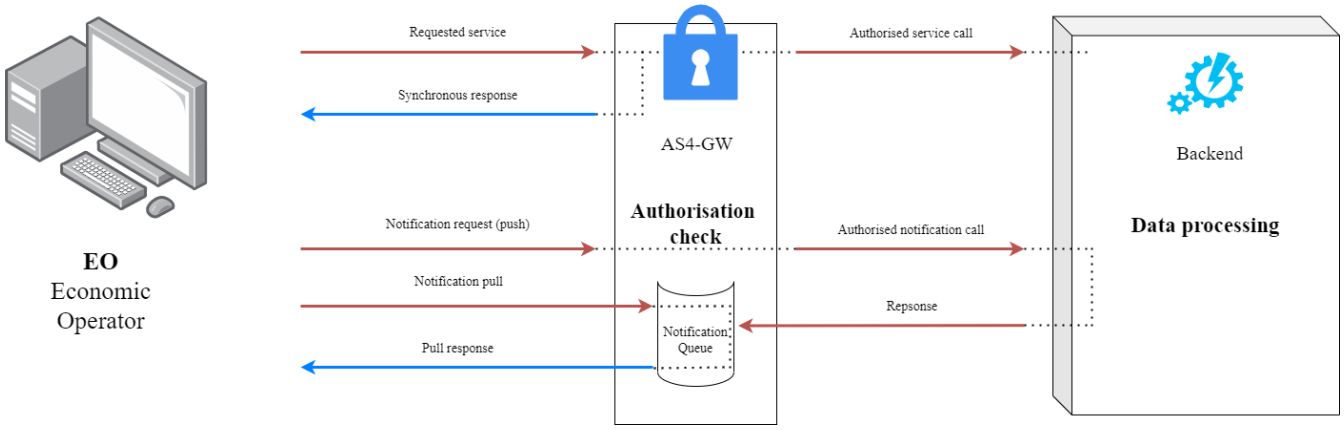

Figure 2.2.1 – System overview

Figure 2.2.1 shows that when lodging a declaration or additional message (requested action), the desired action is called. In the AS4-gateway, the submitted XML is syntax validated, and a synchronous response is returned. In case of a syntax error, the request is synchronously rejected. If the syntax validation is passed, the declaration or additional message is sent to the backend system, where semantic validation and further processing of the declaration or additional message will be carried out.

To know if the declaration or additional message has been accepted or received and what its status is, another action – the notification action – needs to be called by the user. When called with a specific set of parameters, the notification request returns all notifications available matching those request parameters.

2.3 Services and endpoints

All services and actions are reached through the AS4-gateway from the same endpoint, where the AS4-header indicates which service and action to call.
See details on endpoint and AS4-header in the DMS Connectivity Guide.
A description of all services and actions can be found in appendix, section 12.6 - AS4 Services.

The endpoints are:

Environment	Hostname	Port
UFE	secureftpgatewaytest.skat.dk	6384
TFE	secureftpgatewaytest.skat.dk	6384
Prod	secureftpgateway.skat.dk	6384

Table 2.3.1 – Endpoints

This information is combined with details of the company, creating a complete endpoint:

https://secureftpgatewaytest.skat.dk:6384/exchange/CVR_{CVR}_UID_{UID}

Requesting notifications

After a declaration is submitted, it passes through the system in different states that describe where the declaration is in the customs process. The system communicates the state of a declaration through **notifications**.

Receiving notifications is a two-part process in which the user is responsible for interacting with and reacting correctly to the notifications. The two parts are **Pushing notification requests** and **Pulling notification responses**. Notification requests should be pushed with a given time interval and pulls of notification responses from the AS4-gateway queue should be interwoven in between – an example of how to work with notification pushes and pulls is described in section 3.1.

A notification push contains two timestamps, “from” and “to” accordingly, which describe the time interval within which notifications should be retrieved. Upon pushing a notification request, a synchronous response will be elicited by the AS4-gateway in the form of an ‘OK’ response or an error.

When a push is performed successfully, the AS4-gateway will asynchronously retrieve a response from DMS and position it in a queue within the AS4-gateway from which a pull can then be performed. This will result in a synchronous response with the notifications requested in the push.

It is important to continue pulling notifications from the AS4-gateway until the queue is empty, as not all notifications are generated from a request for specific notifications.

The notification holds information on which declaration the notification relates to, the state of the declaration, customs position response in relation to additional message requests, error codes (if rejected), etc.

In section 8.1and 8.2 there is a list and description of each notification type, how to read it from the notification response and other relevant information. For further information on which notifications to expect from the different declarations and additional message flows, see Appendix, section 12.

3.1 Notification Request Design Suggestion

Notifications are requested in time intervals. We recommend requesting all notifications that occurred in the last 7 minutes every 5 minutes, see the simplified example below. With the suggested pattern it is important to keep track of which notifications has already been received, notifications can be uniquely identified by the NotificationSID as described in section 8.1.2

Action #	Time	Push	Pull	Sync Pull Response	Time range	Comment
1	12:00	1			11:53 - 12:00	A request is sent with a 7-minute time range.
2	12:02		1			A pull is performed at 12:02.
3	12:02			1		Notification response is received at 12:02 containing notifications from 11:53 to 12:00
4	12:05	2			11:58 - 12:05	A request is sent with a 7-minute time range.
5	12:07		2			A pull is performed at 12:07.
6	12:07			2		Notification response is received at 12:07 containing notifications from 11:58 to 12:05

Table 3.1.1 – Notification Request Design (seconds are omitted for simplicity)

3.1.1 Requesting notifications

When requesting a set of notifications, the values below can be specified in the message properties. MessageId of the request is contained in the pull response alongside the requested notifications. The MessageId is part of MessageInfo in the AS4-header.

Attribute	Value	Example
submitterId	The submitted ID described in section 6.1.2 in the Connectivity Guide	13116482
dateTo	must be described in YYYY-MM-DDTHH:MI:ss.SSS **in time zone UTC-0**	2022-02-7T13:00:00.000
dateFrom	must be described in YYYY-MM-DDTHH:MI:ss.SSS **in time zone UTC-0**	2022-02-17T13:00:00.000
Lang	The language of the response (must be EN)	EN

An XML example of the MessageProperties is shown below:

```
<eb3:MessageProperties>
<eb3:Property name="submitterId">13116482</eb3:Property>
<eb3:Property name="dateTo">2022-02-7T13:00:00.000</eb3:Property>
<eb3:Property name="dateFrom">2022-02-17T13:00:00.000</eb3:Property >
<eb3:Property name="lang">EN</eb3:Property>
</eb3:MessageProperties>
```

3.1.2 AS4 Notification Pull Request Header

This section explains which information should be contained in the AS4 header when pulling a notification from a Message Partition Channel (MPC).
Message Partition Channel identifies the queue on the AS4-gateway that you will receive messages from. Only one queue per environment exists in DMS, see below.

Environment	Message Partition Channel
Production	urn:fdc:dk.skat.mft.DMS/import2/response
TFE	urn:fdc:dk.skat.mft.DMS/import2/response
UFE	urn:fdc:dk.skat.mft.DMS/import2/response

The following attributes must be provided when pulling:

Attribute	Value	Example
MessageInfo.Timestamp	YYYY-MM-DDTHH:MI:ss.SSSZ	2021-01-19T15:24:37.376Z
MessageInfo.MessageId	GUID	4874bbf7-33c0-49cb-8b98-ca399fccf34a
PullRequest.Property[mpc]	Message Partition Channel	urn:fdc:dk.skat.mft.DMS/import2/response

A full XML example of the messaging header is shown below:

```
<eb3:Messaging xmlns:eb3="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/cor
xmlns:mustUnderstand="http://www.w3.org/2003/05/soap-envelope" xmlns:wsu="http:
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" mustUnderstan
wsu:Id="id-4b28503707cbf61-71e4-43e6-9082-44ff51d30921">
 <eb3:SignalMessage>
  <eb3:MessageInfo>
   <eb3:Timestamp>2022-02-04T05:09:48.099Z</eb3:Timestamp>
   <eb3:MessageId>4874bbf7-33c0-49cb-8b98-ca399fccf34a@skdev00ftpsx.
dmz23.local</eb3:MessageId>
  </eb3:MessageInfo>
  <eb3:PullRequest mpc="urn:fdc:dk.skat.mft.DMS/import2/response"/>
 </eb3:SignalMessage>
</eb3:Messaging>
```

3.1.3 AS4 Notification Pull Response

A notification pull response from the AS4 gateway consist of the SOAP envelope, and, in case the queue is not empty, an additional XML document containing the actual notifications. How to read the additional XML document is described in section 8.1.2. A full response to a pull request can be seen on Danish Customs and Tax Administration's GitHub.

The message property "*RefToOriginalMessageId*" seen below contains the MessageID of the message used to generate the response, such as the MessageID of a request of specific notifications (see 3.1.1.). *RefToOriginalMessageId* should therefore be used to keep track of responses to messages.

It is important to keep track of the specific requests that have been answered, as the system has previously dropped notification requests.

```
<eb:MessageProperties>
    <eb:Property name="RefToOriginalMessageId">a46efff9-e56b-48fb-8439-
627843cce50b</eb:Property>
</eb:MessageProperties>
```

When the queue is empty, the AS4 gateway response contains only the soap envelope. A complete example of a full soap envelope is shown below.

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <eb:Messaging xmlns:eb=http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/
soapenv:mustUnderstand="true">
        <eb:SignalMessage>
            <eb:MessageInfo>
                <eb:Timestamp>2022-06-16T08:13:17.207Z</eb:Timestamp>
                <eb:MessageId>20fc5a2f-8184-40ee-80b4-89825a538f6f@SKAT-MFT</eb:MessageId>
                <eb:RefToMessageId>3c3d0efd-3e33-4749-9315-74a887e31ed5</eb:RefToMessageId>
            </eb:MessageInfo>
            <eb:Error category="Communication" errorCode="EBMS:0006" origin="ebMS"
severity="warning" shortDescription="EmptyMessagePartitionChannel">
                <eb:Description xml:lang="en-US">No message for requested MPC
(urn:fdc:dk.skat.mft.DMS/import2/response)</eb:Description>
<eb:ErrorDetail>//SOAPHeader/eb:Messaging/eb:SignalMessage/eb:PullRequest</eb:ErrorDetail>
            </eb:Error>
        </eb:SignalMessage>
    </eb:Messaging>
</soapenv:Header>
<soapenv:Body/>
</soapenv:Envelope>
```

The error returned for an empty queue **does not indicate a problem** with the communication, but simply that the queue has been emptied. This is **expected behaviour**.

Import declaration submission and additional messages

4

All Import declarations and additional messages can be submitted through DMS Online, as well as through DMS System-to-System in an XML format. In both cases, the data that needs to be provided is defined by the EUCDM standard.

All schemas used in DMS are available on Danish Customs and Tax Administration's GitHub.

Information on how to submit the declarations and the additional messages is described in DMS Connectivity Guide on GitHub.

4.1 Overview of declaration types with their possible additional messages

This section displays an overview of Import declaration types and additional messages. **Note** that some declaration types are additional messages in themselves. For example the I2 Goods Presentation declaration, is a goods presentation message for an H7 declaration.

A short description of the different declarations types is provided in DMS Onboardingguide (in Danish) on GitHub.

Function	H1	H2	H3	H4	H5	H6	H7	I1	I2
Submission	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**X**
Correction	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**-**
Amendment	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**-**
Invalidation	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**-**
Invalidation and Repayment	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**-**
Goods presentation	N/A	N/A	N/A	N/A	N/A	N/A	**X**	N/A	**-**

Table 4.1.1 – Declaration types

N/A indicates that the functionality has not been developed yet.

4.2 Submission

The main functionality is the submission of declarations. Declarations can either be filled out as a standard (IMA), or as a pre-lodged (IMD) declaration. IMA declarations are only sent to DMS and must therefore include a Goods location. An IMD is pre-lodged and goods location must be reported to Manifest and presented to Manifest upon arrival. Presentation of goods can also be done using the I2 Goods Presentation notification. Using I2 requires a customs decision (Bevilling) as Approved Consignee (ACE). Please clarify with the Danish Customs Agency's customer service how to present.

Submissions are sent to the system using the Submission XSD (see section 4.1) for the relevant declaration type. How to fill in the XML schema for submission and which rules to adhere to can be found in the XML guides for the different declaration types under the related declaration type folder.

When submitting a declaration, the **Declaration.Submit** service should be used, see section 12.6.

4.3 Correction

A correction request can be submitted to a declaration **before** the goods have been presented and **before** the declaration has been accepted (when the CWMACC notification is received, see section 8.1, meaning that the declaration still must be in its **pre-lodged** state.

Corrections are sent to the system using the Amendment XSD. It is important that there is at least one changed data element when submitting a correction. If not the correction request will be rejected with the given error code (see section 8.2.6 and section 8.2.18).

Rules and details on how to fill out the data elements in the XML for each declaration type, and which data elements can be corrected, can be found on GitHub under the related declaration type folder. Go to section 4.1 to see which Import declaration types that can be corrected.

When submitting a correction request the **Declaration.Amend** action should be used, see section 12.6.

4.4 Amendment

An amendment request can be submitted to a declaration **after** the goods have been presented and **after** the declaration has been accepted (the CWMACC notification is received, see section 8.2.1), meaning that the declaration must be an **IMA**, either by directly being submitted as a standard declaration or by having had the goods presented.

Amendments are sent to the system using the Amendment XSD. It is important that there is at least one changed data element when submitting an amendment, if not the amendment request will be rejected with the given error code (see section 8.2.6 and section 8.2.18).

Rules and details on how to fill out the data elements in the XML for the different declaration types, and which data elements can be amended, can be found on Github under the related declaration type folder, go to section 4.1 to see which declaration types can be amended.

When an amendment request has been submitted, a customs officer will have to manually grant the request. Therefore, it may take some time before the expected notifications arrive.

When submitting an amendment request the **Declaration.Amend** action should be used, see section 12.6.

4.5 Invalidation

An invalidation request can be submitted to a declaration when needed in case of non-repayment. If there has been a payment of customs debt on the declaration, use the Invalidation and Repayment request (see section 4.6).

Invalidation requests are sent to the system using the Invalidation XSD. A customs officer will have to manually grant or deny the request. Therefore, it might take some time before the expected notifications arrive and the declaration is invalidated.

Rules and details on how to fill out the data elements in the XML for invalidating the different declaration types can be found on GitHub here under the related declaration type folder, go to section 4.1 to see which declaration types can be invalidated.

When submitting an invalidation request the **Declaration.Invalidate** action should be used, see section 12.6.

4.6 Invalidation and Repayment

An invalidation and repayment request can be submitted to a declaration **after** the goods have been presented, **after** the declaration has been accepted (the CWMACC notification is received, see section 8.2.1) and **after** the customs debt has been paid.

Invalidation and repayment requests are sent to the system using the Invalidation and Repayment XSD. A customs officer will have to manually grant the request. Therefore, it might take some time before the expected notifications arrive and the declaration is invalidated.

Rules and details on how to fill out the data elements in the XML for invalidating the different declaration types can be found on GitHub here under the related declaration type folder, go to section 4.1 to see which declaration types an invalidation and repayment request can be submitted to.

When submitting an invalidation and repayment request the **Declaration.InvalidateRemissionRepayment** action should be used, see section 12.6.

4.7 Remission and Repayment

Currently not available as a separate functionality.

4.8 I2 - Goods Presentation

Most EOs will not use this, as the main presentation notification is to be delivered through Manifest.

> **Please implement and use the I2 Presentation notification only if explicitly told so during your onboarding process.**

An I2 Goods Presentation notification can be submitted to a pre-lodged H7 declaration (IMD) when the goods are to be presented. The I2 Goods Presentation notification is used for some specific scenarios but contains similar information as the one delivered to Manifest.

I2 Goods Presentation notifications are sent to the system using the I2 XSD. The Goods Presentation notification follows the I2 EUCDM standard, where details on specific data elements and their usage can be seen in the provided schema here.

When submitting an I2 Goods Presentation notification the **Declaration.Amend.Goodspresented** action should be used, see section 12.6.

Export declaration submission and additional messages

The submission of an export declaration (B1-4/C1-2) starts an export flow of the specified goods, and export flow completes when the declaration is invalidated or when DMS receives an Exit Result from the Office of Exit.

All declarations and additional messages can be submitted through DMS Online, as well as through DMS System-to-System in an XML format. In both cases, the data that needs to be provided is defined by the EUCDM standard. Rules to adhere to can be found in the DMS Export XML guide. All schemas used for DMS Export are available on the Danish Customs and Tax Administration's GitHub.

Information on how to submit the declarations and the additional messages are described in DMS Connectivity Guide on Danish Customs and Tax Administration's GitHub.

5.1 Overview of declaration types with their possible additional messages

This section displays an overview of export declaration types and additional messages.
Note that some declaration types, which are considered declarations customs-wise, are categorized as "additional messages" in the system. For example, the C2 Goods Presentation declaration is a goods presentation additional message for an export declaration and cannot stand alone in the context of the system.

NB!: Correction/amendment (COR), cancellation/invalidation (INV), supplementary declaration (SUP) and C2 goods presentation (GPR) are all considered "additional messages" in the system. These must all refer to an initial declaration via an MRN and are not stand-alone declarations.

A short description of the different declaration types is provided in DMS Onboarding Guide (in Danish) on Danish Customs and Tax Administration's GitHub

Function	B1	B2	B3	B4	C1	C2
Submission	X	X	X	X	X	X
Correction	X	X	X	X	X	-
Amendment	X	X	X	X	X	-
Invalidation	X	X	X	X	X	-
Supplementary declaration	-	-	-	-	X	-
Goods presentation	X	X	X	X	X	-

Table 5.1.1 – Declaration types

5.2 Submission

The submission of an export declaration (B1-4/C1) starts the export flow of the specified goods. Declarations can either be filled in and submitted as a standard declaration (EXA, EXB, EXC,COA), or as a pre-lodged declaration (EXD, EXE, EXF,COD). When submitted, standard declarations are immediately process through the full declaration flow, whereas a pre-lodged declaration is “paused” until the goods are presented with a C2 Presentation Notification.

Submissions are sent to DMS using the Submission XSD (see section 5.1). How to fill in the XML schema for submission and which rules to adhere to can be found in the DMS Export XML guide.

When submitting a declaration, the **Declaration.Submit** action should be used, see section 12.6.

5.3 Correction

A correction is a way of correcting errors and inaccuracies in a **pre-lodged** declaration.

A correction can be submitted to a declaration **before** the goods have been presented and **before** the declaration has been accepted, meaning that the declaration still must be in its **pre-lodged** state.

Rules and details on how to fill out the data elements in the XML for each declaration type, and which data elements can be corrected, can be found in the DMS Export XML guide.

Corrections are sent to the system using the Amendment XSD. It is important that there is at least one changed data element when submitting a correction. If not, the correction request will be rejected with the given error code (see section 8.2.6 and section 8.2.18).

When submitting a correction request the **Declaration.Amend** action should be used, see section 12.6.

5.4 Amendment

An amendment is a way of amending data in a standard declaration.

An amendment request can be submitted to a declaration **after** the goods have been presented and **after** the declaration has been accepted (the CWMACC notification is received, see section 8.2.1), either by directly being submitted as a standard declaration or by having had the goods presented.

Rules and details on how to fill out the data elements in the XML for each declaration type, and which data elements can be amended, can be found in the Export XML guide.

Amendments are sent to the system using the Amendment XSD. It is important that there is at least one changed data element when submitting a correction. If not, the amendment request will be rejected with the given error code (see section 8.2.6 and section 8.2.18).

When submitting a correction request the **Declaration.Amend** action should be used, see section 12.6.

5.5 Cancellation

A cancellation request can be submitted to a pre-lodged declaration if it is no longer relevant.

Cancellation requests are sent to the system using the Invalidation XSD. Rules and details on how to fill out the data elements in an invalidation can be found in the DMS Export XML guide.

Cancellation of a pre-lodged declaration that has not been presented, does not require a customs officer to manually grant or deny the request.

When submitting a cancellation request the **Declaration.Invalidate** action should be used, see section 12.6.

5.6 Invalidation

An invalidation request can be submitted to a declaration if the exportation of goods has been cancelled before the goods of left customs territory.

Invalidation requests are sent to the system using the Invalidation XSD. Rules and details on how to fill out the data elements in an invalidation can be found in the DMS Export XML guide.

Invalidation of a standard declaration or a pre-lodged declaration that has been presented, requires a customs officer to manually grant or deny the request. Therefore, it might take some time before the expected notifications arrive and the declaration is invalidated.

When submitting an invalidation request the **Declaration.Invalidate** action should be used, see section 12.6.

5.7 Supplementary

A Supplementary Declaration converts a Simplified Export declaration (C1) to a full export declaration, B1 or B4.

Rules and details on how to fill out the data elements in a Supplementary Declaration can be found in the DMS Export XML guide.

When submitting an Supplementary Declaration the **Declaration.Amend.Supplement** action should be used, see section 12.6.

5.8 C2 Goods Presentation.

A C2 (Goods Presentation) additional message can be submitted to a pre-lodged export declaration when the goods are to be presented. Note that a C2 *is not considered a declaration* in ERMIS, but instead is considered an additional message just like correction/amendment, invalidation and supplementary declaration messages.

Rules and details on how to fill out the data elements in an C2 Goods Presentation can be found in the DMS Export XML guide.

When submitting an C2 Goods Presentation the **Declaration.Amend.Goodspresented** action should be used, see section 12.6.

Exit declaration submission and additional messages

This section will be updated when the processes are ready.

Transit declaration submission and additional messages

7

This section will be updated when the processes is ready.

Notifications

8.1 List of notifications

8.1.1 Overview of notifications

Here is a list of all the notifications that the system produces, as well as a description of when and how they are issued. The tables below list the notifications and illustrate in which business area the various notifications can be received. For a description of the flow of declarations, see appendix, section 12.

Code	Import	Export	Transit
CWMACC	X	X	N/A
CWMCLE	X	X	N/A
CWMCTL	X	X	N/A
CWMINV	X	X	N/A
CWMRCV	X	X	N/A
CWMREJ	X	X	N/A
CWMREQ	X	X	N/A
CWMRES	X	X	N/A
CWMTAX	X	N/A	N/A
CWMCAS	N/A	X	N/A
CWMDOC	N/A	X	N/A
CWMEOG	N/A	X	N/A
CWMGER	N/A	X	N/A
CWMMAC	N/A	X	N/A
CWMROG	N/A	X	N/A
CWMWTR	N/A	X	N/A
CWMSPM	N/A	N/A	N/A

Table 8.1.1.1 – Notifications to expect for the different business areas

N/A indicates that the functionality has not yet been developed.

Code	Title	Description
CWMACC	Declaration acceptance notification	The submitted declaration has been accepted
CWMCLE	Declaration clearance notification	Procedure is accepted and goods can be released.
CWMCTL	Declaration control notification	The declaration has been selected for control

CWMINV	Declaration invalidation notification	The declaration has been invalidated
CWMRCV	Request receival notification	The submitted request has been received
CWMREJ	Rejection notification	The declaration/request has been rejected/cancelled
CWMREQ	Customs position on request notification	Customs position response on request
CWMRES	Result of request notification	Result of corrections made to the declaration, either by submitter or customs
CWMTAX	Customs debt notification	Notification of details on customs debt with which the declarant or the representative is informed about the details of the customs debt
CWMCAS	Manual Handling State Notification	Notification informing the submitter about the state of a manual work task.
CWMDOC	Document Presentation Notification	Notification telling the submitter that he must present one or more documents, related to the declaration. The notification is also used to remind the submitter about a document that had to be submitted already
CWMEOG	Exit of Goods	Notification informing the submitter about goods exit
CWMGER	Notify Exit Confirmation Reminder	Notification reminding the submitter that Exit Results have not yet been received
CWMMAC	Pending manual decision	Notification informing the submitter that a received declaration is pending manual decision
CWMROG	Release of Goods	Notification informing the submitter that the goods can be released
CWMWTR	Work Task Rejection Notification	Notification informing the submitter that a manual work task is rejected
CWMSPM	Special Procedure Timer Expiration Reminder	Notification informing the submitter that a special procedure timer is expiring

Table 8.1.1.2 – List of notification types

For an overview of the notifications that can be expected for submission, and the additional messages, see Appendix in section 12.

8.1.2 Reading notifications

When requesting notifications from a given time interval, the notifications can arrive in bundles. Each notification bundle is indicated by the <Notifications> ..... </Notifications> tags (notice Notifications is in plural) and can contain multiple notifications indicated by multiple <Notification> ..... </Notification> tags. See the example below:

```
<Notifications>
    <Notification>
        <NotificationEventType>CWMxxx</NotificationEventType>
        <NotificationSID>cca1dd33-2f53-4df8-85ff-d8d1727cf972</NotificationSID>
        <Declaration>
            <MRN>21DKXARQJHQNAHO4R0</MRN>
            <LRN>NOTIFICATION_01</LRN>
            <SubmitterReferenceNumber>NOTIFICATION_01</SubmitterReferenceNumber>
              ..........
        </Declaration>
              ..........
    </Notification>
    <Notification>
        <NotificationEventType>CWMxxx</NotificationEventType>
        <NotificationSID>ea989da5-bf32-4fa7-84ae-a6c02b0a1302</NotificationSID>
        <Declaration>
            <MRN>21DKUYRRHDAKJ512R3</MRN>
            <LRN>NOTIFICATION_02</LRN>
            <SubmitterReferenceNumber>NOTIFICATION_02</SubmitterReferenceNumber>
              ..........
        </Declaration>
              ..........
    </Notification>
    ..........
</Notifications>
```

Table 8.1.2.1 – Notification example

All notifications have **common data elements** that provide information of the declaration. However, the fields and information included after the <SubmitterReferenceNumber>-element in the <Notification>-elements depend on the notification type.

An overview of the information contained in the different **common data elements** can be seen below:

Element name	Description
NotificationEventType	The type of notification, defined as the code described in Table 4.2.1
NotificationSID	A unique ID used for all notifications
Declaration	Contains information that applies to the entire declaration
MRN	The MRN of the submitted declaration that the notification belongs to
LRN	The LRN of the submitted declaration
SubmitterReferenceID	The submitted LRN on the declaration

Table 8.1.2.2 – Information contained in Notifications

The following sections will give an insight into the different notification types, what to be aware of, and how to read them. For a larger overview of the different data elements, and for the notifications they occur in, see Appendix, section 12.

8.2 Notification descriptions

8.2.1 CWMACC - Declaration acceptance notification

The declaration acceptance notification informs the submitter that the declaration has been accepted.

If there are no errors in the declaration, the notification will appear when submitting a standard declaration, or after presenting the goods declared in a pre-lodged declaration.
You can see in the Appendix when CWMACC will appear in the notification flow.

8.2.1.1 Technical description

Below is an example of the CWMACC-notification:

```
<Notification>
    <NotificationEventType>CWMACC</NotificationEventType>
    <NotificationSID>d6cdf594-014d-4015-8e2c-b52488fe6eb5</NotificationSID>
    <Declaration>
        <MRN>21DKYUDDGTIGAYF4R6</MRN>
        <LRN>CWMACCNOTIFICATION</LRN>
        <VersionID>1</VersionID>
        <SubmitterReferenceNumber>CWMACCNOTIFICATION</SubmitterReferenceNumber>
        <AcceptanceDateTime>
            <DateTimeString formatCode="304">20210831073944Z</DateTimeString>
        </AcceptanceDateTime>
         <SubmitterID>12345678</SubmitterID>
    </Declaration>
    <NotificationCreatedDate>
        <DateTimeString formatCode="304">20210831073955Z</DateTimeString>
    </NotificationCreatedDate>
```

```
</Notification>
```

Table 8.2.1.1 – Notification example

A CWMACC notification can in certain circumstances also contain an error or warning message, as seen in the following example:

```
<Notification>
    <NotificationEventType>CWMACC</NotificationEventType>
    <NotificationSID>d6cdf594-014d-4015-8e2c-b52488fe6eb5</NotificationSID>
    <Declaration>
        <MRN>21DKYUDDGTIGAYF4R6</MRN>
        <LRN>CWMACCNOTIFICATION</LRN>
        <VersionID>1</VersionID>
        <SubmitterReferenceNumber>CWMACCNOTIFICATION</SubmitterReferenceNumber>
        <AcceptanceDateTime>
            <DateTimeString formatCode="304">20210831073944Z</DateTimeString>
        </AcceptanceDateTime>
       <SubmitterID>12345678</SubmitterID>
    </Declaration>
<Error>
    <ValidationCode>DKW11607</ValidationCode>
    <Pointer>
        <DocumentSectionCode>
            $.consignmentShipment[?(@.sequenceNumber == 1)].goodsItems[?(@.sequenceNumber == 1]
        </DocumentSectionCode>
    </Pointer>
</Error>
    <NotificationCreatedDate>
        <DateTimeString formatCode="304">20210831073955Z</DateTimeString>
    </NotificationCreatedDate>
</Notification>
```

Table 8.2.1.2 – CWMACC Notification with an error

Contained within the `<Error>` element is the `<ValidationCode>` and `<Pointer>` elements. These indicate the code of the error as well as specifying the erroneous element. In this example, something in the first goods item of the first consignment is erroneous with the validation code `DKW11607`. This error did not result in a rejection of the declaration, but it is best practice to take care to amend the erroneous data when receiving an error in the CWMACC notification.

Besides from the common data elements described in section 8.1.2 there are only a few elements that can be retrieved from this notification:

Element name	Description
VersionID	The version number of the declaration. If corrections or changes (eg. presentation of goods) has been made to the declaration before it has been accepted, this number will be an

	integer >1 depending on how many times changes have been applied
AcceptanceDateTime	The date and time of the acceptance of the declaration
NotificationCreatedDate	The time of creation of the notification. The same as *IssueDateTime*

Table 8.2.1.3 – Information in Notification

8.2.2 CWMCLE - Declaration clearance notification

The CWMCLE notification contains information about the clearance for the procedure, and therefore also about the release of the goods (if this has not already been done). It is sent out only after (though not necessarily directly after) the declaration has been accepted and the CWMACC-notification has been sent.

8.2.2.1 Technical description

```
<Notification>
        <NotificationEventType>CWMCLE</NotificationEventType>
        <NotificationSID>55776a77-686a-4356-bbcc-96076bfed6cc</NotificationSID>
        <Declaration>
            <MRN>21DKI9XIGESJOSWER9</MRN>
            <LRN>CWMCLENOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMCLENOTIFICATION</SubmitterReferenceNumber>
        </Declaration>
        <AdditionalInformation>
            <StatementCode>Considered Satisfactory</StatementCode>
            <StatementTypeCode>AFB</StatementTypeCode>
        </AdditionalInformation>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210825122955Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.2.1 – Notification example

As seen in the sample, the notification contains a section called `<AdditionalInformation> …. </AdditionalInformation>`. The additional information contains information relevant for the trader.

Based on what is indicated in the `<StatementCode>` the additional information can be different types of information. See the table below:

Element name	Description
VersionID	The version of the declaration that has been cleared
AdditionalInformation	Contains relevant information for the submitter

Element name	Description
StatementCode	Description of the relevant information. In the example above it shows the result of the control of the goods
StatementTypeCode	Describes what kind of additional message the Additional Message is – 'AFB' is a Customs Position Motivation.

Table 8.2.2.2 – Information in Notification example

8.2.3 CWMCTL – Control Notification

The CWMCTL notification informs the submitter that the related declaration has been selected for control. Because control has to be performed, it might take longer than usual for the declaration to go through the flow.

8.2.3.1 Technical description

```
<Notification>
        <NotificationEventType>CWMCTL</NotificationEventType>
        <NotificationSID>0ed8fb30-68b2-4f64-bcbe-5bbf938641f9</NotificationSID>
        <Declaration>
            <MRN>22DKCBQY0YHWGDMOR0</MRN>
            <LRN>CWMCTLNOTIFICATION </LRN>
            <SubmitterReferenceNumber>CWMCTLNOTIFICATION</SubmitterReferenceNumber>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20220201135121Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 4.2.3.1 – Notification example

This notification does not contain any information regarding the control. It is only sent to notify the submitter of the control.

8.2.4 CWMINV - Declaration invalidation notification

The CWMINV notification appears when an already accepted declaration has been invalidated**. For the declaration to reach the 'Invalidated' state, a customs officer (in most cases) must approve the invalidation request**. It is the final notification to be sent in the Invalidation flow.

8.2.4.1 Technical description

```
    <Notification>
        <NotificationEventType>CWMINV</NotificationEventType>
        <NotificationSID>6c6a763f-bf9a-4088-93f5-bbdffc2ad226</NotificationSID>
        <Declaration>
```

```
            <MRN>21DKYUDDGTIGAYF4R6</MRN>
            <LRN>CWMINVNOTIFICATION</LRN>
            <SubmitterReferenceNumber>CWMINVNOTIFICATION</SubmitterReferenceNumber>
        </Declaration>
        <AdditionalInformation>
            <StatementCode>3</StatementCode>
            <StatementTypeCode>AFB</StatementTypeCode>
        </AdditionalInformation>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20210912084006Z</DateTimeString>
        </NotificationCreatedDate>
    </Notification>
```

Table 8.2.4.1 – Notification example

The notification provides information on the invalidation request in the <AdditionalInformation>-element

Element name	Description
AdditionalInformation	Contains additional information about the request
StatementCode	Encoded reason for invalidation. In the example **3 is 'invalidation pr trader's request'.**
StatementTypeCode	Describes what kind of additional information the Additional information is – **'AFB' is a Customs Position Motivation**.

Table 8.2.4.2 – Information in Notification example

8.2.5 CWMRCV – Receival notification

The CWMRCV notification informs the trader that their request (an additional message, see section 195.1) was received. Since the request can be many kinds of declaration or additional message, there are ways to distinguish which kind of declaration or additional message request was received. An explanation on how to tell the difference is provided in the following sections:

8.2.5.1 CWMRCV of a pre-lodged declaration

When submitting a pre-lodged declaration, the following CWMRCV-notification will look like the example below:

```
    <Notification>
        <NotificationEventType>CWMRCV</NotificationEventType>
        <NotificationSID>287df338-e51e-440f-bdff-4e3253fbc874</NotificationSID>
        <Declaration>
            <MRN>21DKRSYEMQS5OOTGR1</MRN>
            <LRN>CWMRCVNOTIFICATION_01</LRN>
            <SubmitterReferenceNumber>CWMRCVNOTIFICATION_01</SubmitterReferenceNumber>
             <SubmitterID>12345678</SubmitterID>
        </Declaration>
        <NotificationCreatedDate>
```

```
            <DateTimeString formatCode="304">20210915172430Z</DateTimeString>
        </NotificationCreatedDate>
    </Notification>
```

Table 8.2.5.1 – Notification example

As seen in the example above, there is no further information than the common elements described in section 8.1.2. The CWMRCV notification for a pre-lodged declaration **does not contain** an <AdditionalMessage> element, see section below.

8.2.5.2 CWMRCV of a pre-lodged declaration with warnings

Warnings are sent after submission of a pre-lodged declaration when there is something that the submitter should be aware of, e.g., a quota or restriction on a goods item, or if there are errors in the declaration.

Instead of initially rejecting a pre-lodged declaration with data that would have resulted in a rejection (CWMREJ) of the declaration upon goods presentation, the submitter receives warning codes in the CWMRCV notification. The submitter then has a chance to submit a correction request and thereby correct the erroneous data (see also section 4.3 (Import) and 5.3 (Export)).

```
<Notification>
        <NotificationEventType>CWMRCV</NotificationEventType>
        <NotificationSID>c13210dc-25a1-4e6b-b961-2c01786f2742</NotificationSID>
        <Declaration>
            <MRN>21DKOSUS711H36XJR7</MRN>
            <LRN>CWMRCVNOTIFICATION_02</LRN>
            <SubmitterReferenceNumber>CWMRCVNOTIFICATION_02</SubmitterReferenceNumber>
             <SubmitterID>12345678</SubmitterID>
        </Declaration>
        <Error>
            <ValidationCode>DKW2012</ValidationCode>
        </Error>
        <Error>
            <ValidationCode>DKW2011</ValidationCode>
        </Error>
        <Error>
            <ValidationCode>DKW2005</ValidationCode>
        </Error>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20210916082921Z</DateTimeString>
        </NotificationCreatedDate>
</Notification>
```

Table 8.2.5.2 – Notification example

There can be multiple warnings sent in the CWMRCV notification, all shown in an <Error>-element. The <ValidationCode> element contains the warning code indicating what the error is and thereby which data elements should be corrected.

For a full list of warnings and error codes, see the document Error and Warningcodes found on github.

8.2.5.3 CWMRCV of an additional message

After submission of an additional message (see section 195.1 for more details), the submitter will receive the CWMRCV notification when the message has been received by the system.
To be able to refer to that additional message, an MRN is assigned to it.

```
<Notification>
    <NotificationEventType>CWMRCV</NotificationEventType>
    <NotificationSID>c2159418-5726-4448-8f15-d5a7167d8da2</NotificationSID>
    <Declaration>
        <MRN>21DK6QXM5OVPTWONR2</MRN>
        <LRN>CWMRCVNOTIFICATION_03</LRN>
        <SubmitterReferenceNumber>CWMRCVNOTIFICATION_03</SubmitterReferenceNumber>
    </Declaration>
    <NotificationCreatedDate>
        <DateTimeString formatCode="304">20210916090154Z</DateTimeString>
    </NotificationCreatedDate>
    <AdditionalMessage>
        <MRN>21DKCORFEMO96YDO05</MRN>
    </AdditionalMessage>
</Notification>
```

Table 8.2.5.3 – Notification example

In the example above, the `MRN` of the initial declaration that the request was submitted to can be seen in the top of the notification under the `<Declaration>` element, whereas the `MRN` of the additional message/request is stated in the `<AdditionalMessage>` element.

A way to distinguish which type of request the CWMRCV belongs to is to look at the URN in the <AdditionalMessage> element:

For a CWMRCV notification received from the receival of a **correction/amendment-request**, the URN will be given as
- xxxx**COR**xxxxxxxxxxx – the 5th to 7th characters is '**COR**'

For a CWMRCV notification received from the receival of an **invalidation request**, the URN will be given as
- xxxx**INV**xxxxxxxxxxx – the 5th to 7th characters are '**INV**'

For a CWMRCV notification received from the receival of an **I2/C2/Goods Presentation Notification**, the URN will be given as
- xxxx**GPR**xxxxxxxxxxx – the 5th to 7th characters are '**GPR**'

For a CWMRCV notification resulting from the receival of a **Supplementary declaration for a C1 (Simplified Declaration),** the URN will be given as
- xxxx**SUP**xxxxxxxxxxx – the 5th to 7th characters are '**SUP**'

The URN of the additional message will also appear in the CWMREJ notification under the `<MRN>`-element if the additional message is rejected, and in the CWMREQ notification when Customs has taken position on the additional message (see more in section 8.2.6 and 8.2.7). Thereby, it can be seen which received additional message has been rejected or taken position on.

8.2.5.4 Specific for DMS Transit

This section will be updated when the processes is ready.

8.2.6 CWMREJ - Rejection notification

8.2.6.1 CWMREJ for a declaration

Receiving a CWMREJ notification after submitting a declaration means that there are errors in the submitted declaration that result in the declaration not passing validation. Whether it is breaking a business rule or submitting an invalid code or ID, the CWMREJ notification contains information on the specifics of the error(s).

```
    <Notification>
        <NotificationEventType>CWMREJ</NotificationEventType>
        <NotificationSID>4dcdf061-a884-4fde-a749-91adeddb492f</NotificationSID>
        <Declaration>
            <LRN>CWMREJNOTIFICATION_01</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMREJNOTIFICATION_01</SubmitterReferenceNumber>
            <RejectionDateTime>
                <DateTimeString formatCode="304">20210916080304Z</DateTimeString>
            </RejectionDateTime>
             <SubmitterID>12345678</SubmitterID>
        </Declaration>
        <Error>
            <ValidationCode>DK1045</ValidationCode>
            <ValidationText>Error in "Importer Identification no." (3/16), when "Additional proce-
dure" (1/11) is F49, the "Importer Identification no" (3/16) must be DK09999981.</ValidationText>
            <Pointer>
                <DocumentSectionCode>$.consignmentShipment[?(@.sequenceNumber ==0)].par-
ties[?(@.partyRoleType == IM) ].partyIdentification</DocumentSectionCode>
            </Pointer>
            <Pointer>
                <DocumentSectionCode>$.consignmentShipment[?(@.se-
quenceNumber == 0)].goodsItems[?(@.sequenceNumber ==1)].procedureCombination</DocumentSectionCode>
            </Pointer>
        </Error>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20210916080305Z</DateTimeString>
        </NotificationCreatedDate>
    </Notification>
```

Table 8.2.6.1 – Notification example

As with the CWMRCV notification with warnings, the CWMREJ also includes information on the error in the `<Error>`-element. However, there are a few more elements containing information on where to find the error:

In the example above a business rule was broken. The broken business rule is explained as an error code in `<ValidationCode>` and error description in `<ValidationText>`. The `<Pointer>` -element(s) indicate(s) which data element(s) in the declaration should be changed for the declaration to be accepted.

For a full list of warnings and error codes, see the document Error and Warningcodes found on github.

8.2.6.2 CWMREJ of an additional message

Faulty additional messages can also be rejected. As with the CWMREJ notification of a rejected declaration, the errors for the rejection of an additional message are also displayed in the <Error> element of the notification.

In the <Error>-element, an error code is contained in the <ValidationCode> element as well as an error description in the <ValidationText> element, as seen in the example below.

```
<Notification>
        <NotificationEventType>CWMREJ</NotificationEventType>
        <NotificationSID>ea596738-a493-451e-86d4-f0c66d2e639f</NotificationSID>
        <Declaration>
            <MRN>21DK6QXM5OVPTWONR2</MRN>
            <LRN>CWMREJNOTIFICATION_02</LRN>
            <SubmitterReferenceNumber>CWMREJNOTIFICATION_02</SubmitterReferenceNumber>
            <RejectionDateTime>
                <DateTimeString formatCode="304">20210916081418Z</DateTimeString>
            </RejectionDateTime>
        </Declaration>
        <Error>
            <ValidationCode>DMS10001</ValidationCode>
            <ValidationText>Obligation error: obligation rule not met</ValidationText>
        </Error>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20210916081419Z</DateTimeString>
        </NotificationCreatedDate>
        <AdditionalMessage>
            <MRN>21DKCORFEMO96YDO05</MRN>
        </AdditionalMessage>
    </Notification>
```

Table 8.2.6.2 – Notification example

Unlike the CWMREJ notification that is sent *when a declaration is rejected*, The CWMREJ notification sent *when an additional message is rejected* contains an <AdditionalMessage>-element, **which matches the MRN in the <AdditionalMessage>-element from the CWMRCV notification of the submitted additional message** (see also section 8.2.5.3). The submitter can then know which additional message that has been received and thereafter rejected.

For a full list of warnings and error codes, see the document Error and Warningcodes found on github.

8.2.6.3 CWMREJ after I2/C2 – Goods presentation

The CWMREJ notification can appear after submission of an I2/C2 in two scenarios:

- The I2/C2 additional message is rejected
- The pre-lodged declaration is rejected

Rejection of the I2/C2 additional message

In case the I2/C2 additional message is rejected, the CWMREJ notification will contain error code(s) and error description(s) in an <Error> element describing the error(s) present. It will also contain an <AdditionalMessage>-element

indicating that it is the additional message that is rejected and not the initial pre-lodged declaration. The MRN in the <AdditionalMessage>-element will be on the format xxxx**GPR**xxxxxxxxxxx, meaning that the additional message that is being rejected is a GPR – Goods Presentation. See also the example below.

```
<Notification>
        <NotificationEventType>CWMREJ</NotificationEventType>
        <NotificationSID>590f9ad6-b9f5-4abf-bec7-3837a219fe30</NotificationSID>
        <Declaration>
            <MRN>21DKIMEZICQN4VJJR2</MRN>
            <LRN>CWMREJNOTIFICATION_03</LRN>
            <SubmitterReferenceNumber>CWMREJNOTIFICATION_03</SubmitterReferenceNumber>
            <RejectionDateTime>
                <DateTimeString formatCode="304">20211118102017Z</DateTimeString>
            </RejectionDateTime>
        </Declaration>
        <Error>
            <ValidationCode>DK1072</ValidationCode>
            <Pointer>
                <DocumentSectionCode>$.consignmentShipment[?(@.sequenceNumber==0)].loca-
tions[?(@.locationRoleType=="14")].physicalAddress.countryCode</DocumentSectionCode>
            </Pointer>
        </Error>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20211118102019Z</DateTimeString>
        </NotificationCreatedDate>
        <AdditionalMessage>
            <MRN>21DKGPRJBYGXSEVD00</MRN>
        </AdditionalMessage>
    </Notification>
```

Table 8.2.6.3 – Notification example

In this example, the CountryCode element under GoodsLocation was invalid in the I2/C2 additional message. The pre-lodged declaration remains in the state of 'Pending Goods Presentation' and a new (and corrected) I2/C2 additional message can be submitted to present the goods for the initial declaration.

Rejection of the initial pre-lodged declaration

The second case of receiving a CWMREJ notification after submitting a Goods Presentation notification is when the initial pre-lodged declaration ends up being rejected. This can happen when the declaration contains errors, that were given as warnings in its CWMRCV notification (see section 8.2.5.2) which needed to be corrected (but hasn't been corrected) by submitting a correction (see section 4.3 (Import) and 5.3 (Export)).

In this case, the CWMREJ notification will contain the corresponding error code(s) and error description(s) of the warning code(s) from the CWMRCV notification in an <Error>-element describing the error(s) in the declaration. Unlike the CWMREJ notification from the rejection of an I2/C2 additional message, this scenario **does not contain an** <AdditionalMessage> **element**. See example below.

```
<Notification>
```

```
        <NotificationEventType>CWMREJ</NotificationEventType>
        <NotificationSID>c028b808-2989-4610-bc99-388cb120a40b</NotificationSID>
        <Declaration>
            <MRN>21DKH9EYOCY6AGJRR8</MRN>
            <LRN>CWMREJNOTIFICATION_04</LRN>
            <VersionID>2</VersionID>
            <SubmitterReferenceNumber>CWMREJNOTIFICATION_04</SubmitterReferenceNumber>
            <RejectionDateTime>
                <DateTimeString formatCode="304">20211118104422Z</DateTimeString>
            </RejectionDateTime>
        </Declaration>
        <Error>
            <ValidationCode>DK2011</ValidationCode>
            <ValidationText>Error, in "Declarant identification No." (3/18), the declar-
ant must be registered as an importer in DK.</ValidationText>
        </Error>
        <Error>
            <ValidationCode>DK2005</ValidationCode>
            <ValidationText>Error in "Declarant identification No." 3/18, the number does not ex-
ist or is not valid.</ValidationText>
        </Error>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20211118104423Z</DateTimeString>
        </NotificationCreatedDate>
    </Notification>
```

Table 8.2.6.4 – Notification example

In the example above an invalid Declarant ID was declared in the declaration and it was not corrected after receiving warnings in its CWMRCV notification. This results in the declaration being rejected when the goods were presented. In this case, the pre-lodged declaration will have to be resubmitted with a new LRN.

8.2.6.4 Specific for DMS Transit

This section will be updated when the processes is ready.

8.2.7 CWMREQ – Customs position on request notification

When submitting an additional message, the submitter will receive a CWMREQ notification when the additional message has been processed, either by the system or a Customs officer.

8.2.7.1 Technical description

```
<Notification>
        <NotificationEventType>CWMREQ</NotificationEventType>
        <NotificationSID>9bad72db-c2f3-4966-8419-029120c19058</NotificationSID>
        <Declaration>
            <MRN>21DKRSYEMQS5OOTGR1</MRN>
```

```
            <LRN>CWMREQNOTIFICATION</LRN>
            <SubmitterReferenceNumber>CWMREQNOTIFICATION</SubmitterReferenceNumber>
        </Declaration>
        <AdditionalInformation>
            <StatementTypeCode>AFB</StatementTypeCode>
            <StatementDescription>Granted automatically.</StatementDescription>
        </AdditionalInformation>
        <CustomsPosition>
            <ID>1d249c92-4f74-4506-8bec-314232194725</ID>
            <Type>GRANTED</Type>
        </CustomsPosition>
        <NotificationCreatedDate>
            <DateTimeString formatCode="304">20210915172540Z</DateTimeString>
        </NotificationCreatedDate>
        <AdditionalMessage>
            <URN>21DKCORQLZX8POKV08</URN>
        </AdditionalMessage>
    </Notification>
```

Table 8.2.7.1 – Notification example

Note that the CWMREQ notification uses the element `<URN>` Unique Reference Number (URN) to refer to the additional message. The `URN` and `MRN` values are functionally equivalent, so the only difference is the name of the element, as can be verified by checking that the `MRN` field in the CWMRCV matches the `URN` field in the CWMREQ for the additional message in question. This naming may change to be unified as the system is updated.

The notification contains information about the customs position in the `<CustomsPosition>` element, including the ID of the decision as well as the type indicating whether the request was `GRANTED` or `DENIED`.

If there are comments from the customs office, they are included in the `<AdditionalInformation> … </AdditionalInformation>` element.

Note that the `<AdditionalMessage>` element here refers to the reference number of the correction additional message by `<URN>` (Unique Reference Number) as opposed to the CWMRCV notification which uses `<MRN>` to refer to the reference number of the additional message (see section 8.2.5.3). The numbers will be the same, only the name of the data element is different.

Additionally, as with the CWMREJ notification (see section 8.2.6), the notification contains the MRN of both the initial declaration, as well as the MRN of the **additional message** in the `<AdditionalMessage>` element. The MRN of the **additional message** also matches the MRN **of the** `<AdditionalMessage>` **element from the CWMRCV notification of the submitted additional message.**

8.2.8 CWMRES - Result of request notification

The CWMRES notification arrives after the declaration has been accepted (the CWMACC notification is received), and only in cases where the goods are presented after lodging the declaration, or when the declaration is amended after acceptance. It notifies the trader of the result of their request.

When submitting a pre-lodged declaration, the CWMRES notification will contain information on all changes the declaration has gone through in the process, i.e., changing type (from Pre-lodged to Standard) when goods are presented, as well as any changes in or amendment of location of goods or other data elements.

8.2.8.1 Specific for DMS Transit

This section will be updated when the processes is ready.

8.2.8.2 Technical description

A step-for-step explanation of how to read this XML example is included after the example.

```
 <Notification>
        <NotificationEventType>CWMRES</NotificationEventType>
        <NotificationSID>ab73bd28-3417-4653-8d86-8b4ec4f65b9e</NotificationSID>
        <Declaration>
            <MRN>21DKJYMHJEYP3CQDR2</MRN>
            <LRN>CWMRESNOTIFICATION</LRN>
            <VersionID>3</VersionID>
            <SubmitterReferenceNumber>CWMRESNOTIFICATION</SubmitterReferenceNumber>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>1</sequenceNumber>
                <value>24.00</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].goodsItems[?(@.se-
quenceNumber == 1)].declaredCustomsValue.value</pointer>
                <timestamp/>
                <declarationVersion>1</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>2</sequenceNumber>
                <value>A</value>
                <pointer>$.type</pointer>
                <declarationVersion>2</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>3</sequenceNumber>
                <value>DKFDH</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].locations[?(@.locationRole-
Type == '14')].locationId</pointer>
                <timestamp/>
                <declarationVersion>2</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>4</sequenceNumber>
                <value>0003</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].locations[?(@.locationRole-
Type == '14')].locationAdditionalId</pointer>
```

```
                <timestamp/>
                <declarationVersion>2</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>5</sequenceNumber>
                <value>U</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].locations[?(@.locationRole-
Type == '14')].locationIdentificationType</pointer>
                <timestamp/>
                <declarationVersion>2</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>6</sequenceNumber>
                <value>A</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].locations[?(@.locationRole-
Type == '14')].locationType</pointer>
                <timestamp/>
                <declarationVersion>2</declarationVersion>
            </amendment>
            <amendment>
                <createdBy>CWM</createdBy>
                <sequenceNumber>7</sequenceNumber>
                <value>DK</value>
                <pointer>$.consignmentShipment[?(@.sequenceNumber == 0)].locations[?(@.locationRole-
Type == '14')].physicalAddress.countryCode</pointer>
                <timestamp/>
                <declarationVersion>2</declarationVersion>
            </amendment>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210922115010Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.8.1 – Notification example

The <amendment> element gives insight into which changes the declaration has gone through in the process.

In the example above, the pre-lodged declaration has gone through a correction before goods presentation, incrementing the declaration <VersionID> from version 1 to 2. Thereafter, the goods have been presented, and the location of the goods have been added to the declaration during goods presentation, taking the declaration <VersionID> from 2 to 3. This is why the <VersionID> starts with 3 in the CWMRES notification in the top of the example, as this is the current version of the declaration at the time of receiving that notification.

Even if the CWMRES notification arrives after a Goods Presentation is sent to the system, there is no <AdditionalMessage> element, as the changes stated in the CWMRES notification relate to the initial pre-lodged declaration.

The <VersionID> element refers to the current version of the declaration at the time of submitting the additional message. The <declarationVersion> element,

on the other hand, refers to the version of the declaration that the changes apply to. This is why they are different in the example above. The example can be read as: the current version of the declaration is version 3. The change from version 1 to 2 occurred when a correction was submitted, changing the declaredCustomsValue to 24. The change from version 2 to 3 occurred upon presenting the goods, as indicated by the $.type value changing to A for EXA instead of EXD (see example **Table 8.2.8.2** below). Along with this goods presentation, the location of the goods was also added, resulting in amendments for the 5 data elements under the `.locations` pointer.

Note that each change of an individual data element **does not** increment the declaration version. The declaration version is only incremented upon the system receiving an additional message for that declaration, no matter how many changes the additional message results in.

As mentioned in the previous paragraph, after submitting a Goods Presentation (GPR) additional message, the resulting CWMRES notification will always have an `<amendment>` element of the following form:

```
<amendment>
	<createdBy>CWM</createdBy>
	<sequenceNumber>1</sequenceNumber>
	<value>A</value>
	<pointer>$.type</pointer>
	<declarationVersion>1</declarationVersion>
</amendment>
```

Table 8.2.8.2 – Goods presentation CWMRES example

This `<amendment>` element indicates, in the `<pointer>` element, that the declaration has changed `type` – in this case from pre-lodged (type: D) to standard (type: A). When submitting an amendment/correction, the `<pointer>` sub-element will instead contain the pointer to the data element(s) that have been changed by the correction/amendment. For more information on amendments, see section 4.4 (Import) and 5.4 (Export).

Element name	Description
Amendment	
createdBy	The system that created the amendment
sequenceNumber	Number uniquely identifying the amendment object
Value	The updated value of the amended/corrected data element
Pointer	Pointer indication the amended/corrected data element
declarationVersion	The version number of the declaration that was amended/corrected

Table 8.2.8.3 – Information in Notification

8.2.9 CWMROG – Release of Goods

8.2.9.1 Specific for DMS Import

In DMS Import, the CWMROG notification means the exact same thing as the CWMCLE notification and contains information about the release of the goods. It is sent out only after (though not necessarily directly after) the declaration has been accepted and the CWMACC notification has been sent.

This notification is identical to the CWMCLE notification and is considered a bug in the system for DMS Import. It does not mean that an error has occurred, and it should be handled exactly as the CWMCLE notification. It is expected behavior in DMS Export.

8.2.9.2 Specific for DMS Export

In DMS Export, the CWMROG notification follows the CWMCLE notification after an anticipated export record (AER) has been created for the consignment in question. Note that the CWMROG notification is expected in DMS Export, as it notifies the trader that the goods are cleared and ready for release, thus marking the last step before exit procedures begin.

8.2.9.3 Technical description

```
<Notification>
        <NotificationEventType>CWMROG</NotificationEventType>
        <NotificationSID>55776a77-686a-4356-bbcc-96076bfed6cc</NotificationSID>
        <Declaration>
            <MRN>21DKZ5YSSN6JHMQBR3</MRN>
            <LRN>CWMROGNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMROGNOTIFICATION</SubmitterReferenceNumber>
        </Declaration>
        <AdditionalInformation>
            <StatementCode>A1</StatementCode>
            <StatementTypeCode>AFB</StatementTypeCode>
        </AdditionalInformation>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210825122955Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.9.1 – Notification example

As seen in the sample, the notification contains a section called `<AdditionalInformation> …. </AdditionalInformation>`. The additional information contains information relevant for the trader. Based on what is indicated in the `<StatementCode>`, the additional information can be different types of information. See the table below (or the CWMCLE notification, see section 8.2.2):

Element name	Description

VersionID	The version of the declaration that has been cleared
AdditionalInformation	Contains relevant information for the submitter
StatementCode	Description of the relevant information. In the example above it indicated the result of the control of the goods – 'A1' means 'Considered Satisfactory' (as for the CWMCLE notification)
StatementTypeCode	Describes what kind of additional message the Additional Message is – 'AFB' is a Customs Position Motivation.

Table 8.2.9.2 – Information in Notification

8.2.10 CWMTAX - Customs debt notification

> **Note: This notification is only relevant for DMS Import**

8.2.10.1 Specific for DMS Import

When submitting a declaration, the submitter is notified by the CWMTAX notification that a calculation of customs debt has been done. The CWMTAX notification appears after the initial submission of a declaration, which for a standard (IMA) declaration is the final calculation (unless amendments are requested and granted for the customs value of a goods item), and for a pre-lodged (IMD) declaration it is only a preliminary calculation – there will be a recalculated customs debt when goods are presented, and the CWMTAX notification will be sent from the system again.

8.2.10.2 Technical description

```
<Notification>
        <NotificationEventType>CWMTAX</NotificationEventType>
        <NotificationSID>685eefec-f413-425d-a055-927856d36993</NotificationSID>
        <Declaration>
            <MRN>21DKRSYEMQS5OOTGR1</MRN>
            <LRN>CWMTAXNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMTAXNOTIFICATION</SubmitterReferenceNumber>
            <DutyTaxFee>
                <Payment>
                    <ReferenceID>DK19552101:1</ReferenceID>
                    <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                    <TaxAssessedAmount>0</TaxAssessedAmount>
                </Payment>
```

```
            </DutyTaxFee>
            <GoodsShipment>
                <GovernmentAgencyGoodsItem>
                    <SequenceNumeric>1</SequenceNumeric>
                    <Commodity>
                        <DutyTaxFee>
                            <Payment>
                                <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                                <TaxAssessedAmount currencyID="DKK">75.1</TaxAssessedAmount>
                            </Payment>
                            <SpecificTaxBaseQuantity unitCode="DKK">301</SpecificTaxBaseQuantity>
                            <DeductAmount currencyID="DKK">0</DeductAmount>
                            <TaxRateNumeric>25.0</TaxRateNumeric>
                            <TypeCode>B00</TypeCode>
                        </DutyTaxFee>
                    </Commodity>
                </GovernmentAgencyGoodsItem>
            </GoodsShipment>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210915172600Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.10.1 – Notification example

As seen in the example above, the notification contains information of the payment under the `<Declaration>` element and the `<GovernmentAgencyGoodsItem>` element, each having slightly different sub-elements.

Element name	Description
Declaration	Information on the payment for the entire declaration
DutyTaxFee	DutyTaxFee captures Duty/Tax/Fee data of a particular duty/tax/fee type
Payment	This element contains information on a given payment
ReferenceID	Payment ID
PaymentAmount	The actual amount paid, or to be paid, for all items in the declaration, rounded down to one digit.
TaxAssessedAmount	Assessed amount of duty/tax/fee (includes all types of charges and duties). Assessed per duty/tax/fee type by declaration.
GoodsShipment	GoodsShipment captures the data of the shipment of the goods belonging to one particular consignment crossing the border of the Customs area

Element name	Description
GovernmentAgencyGoodsItem	Information on the payment for the specific goods item
SequenceNumeric	The number of the goods item as given on the submitted declaration.
Commodity	Details about the properties of the goods
DutyTaxFee	DutyTaxFee captures Duty/Tax/Fee data of a particular duty/tax/fee type
Payment	This element contains information on the base of the calculation of a given payment
PaymentAmount	The actual amount paid, or to be paid, for the specific item, rounded down to one digit.
TaxAssessedAmount	Assessed amount of duty/tax/fee (includes all types of charges and duties). Assessed per duty/tax/fee type by item
SpecificTaxBaseQuantity	The quantity on which a duty or tax or fee will be assessed (FreightChargeAmount + CustomsValueAmount)
DeductAmount	Amount of relief applicable from a duty or tax
TaxRateNumeric	Rate of duty or tax or fee applicable to commodities or of tax applicable to services (25.00 = 25%)
TypeCode	Code for type of tax to be applied (eg., B00 is VAT)
IssueDateTime	The time of creation of the notification. The same as *NotificationCreatedDate*

Table 8.2.10.2 – Information in Notification

As seen in the example above, the notification contains information of the payment under the <Declaration>-

8.2.11 CWMCAS - Manual Handling State Notification

The manual handling state notification informs the submitter that a manual work task has been created and its state.

In the *Queued for handling* state is the system is awaiting a decision on the declaration or an additional message by a customs officer.

In the *Completed* state the manual decision has been closed, and another notification about the new status of declaration is sent such as a CWMINV if an invalidation was requested, or a CWMREQ in other circumstances (see section 8.2.7).

8.2.11.1 Technical description

Below is an example of the CWMCAS-notification:

```
    <Notification>
        <NotificationEventType>CWMCAS</NotificationEventType>
        <NotificationSID>04246884-0b30-4c32-a751-b9ada11e2298</NotificationSID>
        <Declaration>
            <MRN>22DKBQDUWGOU6DTEA0</MRN>
            <LRN> CWMCASNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMCASNOTIFICATION</SubmitterReferenceNumber>
            <SubmitterID>12345678</SubmitterID>
        </Declaration>
        <AdditionalInformation>
            <StatementCode>41</StatementCode>
            <StatementDescription>Declaration queued for manual case handling</StatementDescrip-
tion>
        </AdditionalInformation>
        <IssueDateTime>
            <DateTimeString formatCode="304">20221012103429Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.11.1 – Notification example

Besides from the common data elements described in section 8.1.2 there are only a few elements that can be retrieved from this notification:

Element name	Description
AdditionalInformation/StatementCode	Provides additional information on the state of the manual task. 41 - Queued for handling 42 - Handling in progress 43 - Completed
AdditionalInformation/ StatementDescription	Provides additional information on the state of the manual task in an easily readable format.

Table 4.2.11.2 – Information in Notification

8.2.12 CWMDOC - Document Presentation Notification

Note: This section is only a preliminary description and may be subject to change as the feature is fully tested.

Notification informing the submitter that one or more documents must be presented in order for the declaration to be valid. The notification may also be used retroactively, that is, to remind the submitter about a document that had to be submitted already.

8.2.12.1 Technical description

Below is an example of the CWMDOC-notification:

```
<Notification>
    <NotificationEventType>CWMDOC</NotificationEventType>
    <NotificationSID>f69e0829-8559-48d4-9c3d-3a9981af9fe6</NotificationSID>
    <IssueDateTime>
        <DateTimeString formatCode="304">2022</DateTimeString>
    </IssueDateTime>
    <DueDate>2022</DueDate>
    <AdditionalDocument>
        <Type>380</Type>
        <Identifier>token</Identifier>
    </AdditionalDocument>
    <AdditionalDocument>
        <Type>A004</Type>
        <Identifier>SUPDOCID123</Identifier>
    </AdditionalDocument>
    <AdditionalDocument>
        <Type>A004</Type>
        <Identifier>SUPDOCID123</Identifier>
    </AdditionalDocument>
    <Control>
        <TypeCode>10</TypeCode>
    </Control>
    <Declaration>
        <SubmitterReferenceNumber> CWMDOCNOTIFICATION</SubmitterReferenceNumber>
        <MRN>22DKL1UDY2XA5YVMA9</MRN>
        <VersionNumber>1</VersionNumber>
    </Declaration>
</Notification>
```

Table 8.2.12.1 – Notification example

Besides from the common data elements described in section 8.1.2 there are only a few elements that can be retrieved from this notification:

Element name	Description
AdditionalDocument\Type	The type of document that must be present or should have been submitted
AdditionalDocument\Identifier	The identifier of document that must be present or should have been submitted
Control\TypeCode	The type of control to be performed on the documents. 10 is documents control.

Table 4.2.12.2 – Information in Notification

8.2.13 CWMEOG - Exit of Goods

8.2.13.1 Specific for DMS Export

Note: This section is only a preliminary description and may be subject to change as the feature is fully tested.

The CWMEOG notification informs the trader of the Exit of Goods for the relevant consignment. After three submissions from the trader to the Office of Exit have been sent, this notification should be sent to the trader. These three submissions are: a Submission of arrival notification (IE507), a Submission of Export Manifest (IE547), and a Submission of Exit Notification (IE590).

8.2.13.2 Specific for DMS Transit

This section will be updated when the processes is ready.

8.2.13.3 Technical description

An example of a CWMEOG notification will be added when the feature is ready.

8.2.14 CWMGER - Notify Exit Confirmation Reminder

Note: This section is only a preliminary description and may be subject to change as the feature is fully tested.

The CWMGER notification is sent to the trader by the system when the time limit to receive exit results has run out, i.e. when the trader has not sent the exit results message to the system in time.

8.2.14.1 Technical description

Below is an example of the CWMGER notification:

```
<Notification>
    <NotificationSID>8b054eee-ca4e-43e7-af59-8eb0d6bfe083</NotificationSID>
    <NotificationEventType>CWMGER</NotificationEventType>
    <IssueDateTime>
        <DateTimeString formatCode="304">2022-10-10T12:39:10.792</DateTimeString>
    </IssueDateTime>
    <Declaration>
        <VersionID>1</VersionID>
        <EffectiveDateTime>[2022,10,10,12,39,10,792752720])</EffectiveDateTime>
        <FunctionalReferenceID>EM17</FunctionalReferenceID>
        <ID>22DK12PWICTBCVN7A2</ID>
        <CustomsOfficeOfExport>DK004700</CustomsOfficeOfExport>
        <Declarant>
            <ID>DK12345678</ID>
            <Contact></Contact>
        </Declarant>
        <Exporter>
            <ID>DK12345678</ID>
        </Exporter>
    </Declaration>
    <Control>
        <SequenceNumeric>1</SequenceNumeric>
```

```
        <LimitDateTime>[2022,12,9,12,39,10,795100117]</LimitDateTime>
    </Control>
</Notification>
```

Table 8.2.14.1 – Notification example

Besides from the common data elements described in section 8.1.2 there are only a few elements that can be retrieved from this notification:

Element name	Description
Declaration	
Control/SequenceNumeric	Numeric identifier for the control
Control/LimitDateTime	The time limit that was exceeded

Table 8.2.14.2 – Information in Notification

8.2.15 CWMMAC - Pending manual decision

8.2.15.1 Specific for DMS Export

Note: This section will be expanded when this functionality has been fully implemented and tested.

CWMMAC is a notification informing the submitter that a received declaration is pending a manual decision before the goods can be released for export.

A CWMMAC notification will occur after a declaration is accepted (CWMACC), but before it is cleared for release (CWMCLE). Therefore, the consignment goods must be presented before a CWMMAC will occur in the case of a pre-lodged declaration. Along with the CWMMAC notification, a CWMCAS notification (see section 8.2.11) will also be sent to the trader, notifying the trader that the case is pending manual handling.

A customs office agent must then manually grant or deny release for the declaration in question. Two additional CWMCAS notifications are sent during this flow – one when the case handling begins, and one when the case handling is completed.

8.2.15.2 Technical description

Below is an example of the CWMMAC-notification:

```
    <Notification>
        <NotificationEventType>CWMMAC</NotificationEventType>
        <NotificationSID>c636b38c-80ea-43bd-8d09-815523074469</NotificationSID>
        <Declaration>
            <MRN>22DKVGUXWP4XUDHKA4</MRN>
            <LRN>CWMMACNOTIFICATION</LRN>
            <SubmitterReferenceNumber>CWMMACNOTIFICATION</SubmitterReferenceNumber>
            <SubmitterID>12345678</SubmitterID>
        </Declaration>
```

```
        <AdditionalInformation>
            <StatementCode>INF - Supporting Document Type has value C605 or C700</StatementCode>
            <StatementDescription>MANUAL_RELEASE</StatementDescription>
        </AdditionalInformation>
        <IssueDateTime>
            <DateTimeString formatCode="304">20221020123418Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.15.1 – Notification example

Besides from the common data elements described in section 8.1.2 there are only a few elements that can be retrieved from this notification:

Element name	Description
AdditionalInformation	
StatementCode	Code describing the reason for manual release decision
StatementDescription	Description of the manual release decision

Table 8.2.15.2 – Information in Notification

8.2.16 CWMWTR - Work Task Rejection Notification

8.2.16.1 Specific for DMS Export

Note: This section will be expanded when this functionality has been fully implemented and tested.

The CWMWTR notification is triggered when a customs agent denies the manual decision for the release of goods as notified by the CWMMAC notification. It notifies the trader that the request for release has been rejected.

8.2.16.2 Technical description

Below is an example of the CWMWTR-notification:

```
    <Notification>
        <NotificationEventType>CWMWTR</NotificationEventType>
        <NotificationSID>f32dd1c1-bbf9-44bf-9be2-3f8a39cc746f</NotificationSID>
        <Declaration>
            <MRN>22DK66NYEVJRSHEHA2</MRN>
            <LRN>CWMWTRNOTIFICATION</LRN>
            <SubmitterReferenceNumber>CWMWTRNOTIFICATION</SubmitterReferenceNumber>
            <SubmitterID>12345678</SubmitterID>
        </Declaration>
        <AdditionalInformation>
            <StatementCode>INF - Supporting Document Type has value C605 or C700</StatementCode>
            <StatementTypeCode>MANUAL_RELEASE</StatementTypeCode>
            <StatementDescription>null</StatementDescription>
        </AdditionalInformation>
```

```
        <IssueDateTime>
            <DateTimeString formatCode="304">20221020130846Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.17.1 – Notification example

Besides from the common data elements described in section 8.1.2 the following elements can be retrieved from the notification:

Element name	Description
AdditionalInformation	
StatementCode	Code describing the reason for manual release decision
StatementTypeCode	TypeCode for the notification
StatementDescription	Description of the manual release decision

Table 8.2.17.2 – Information in Notification

8.2.17 CWMSPM - Special Procedure Timer Expiration Reminder

8.2.17.1 Specific for DMS Export

Note: This section will be expanded when this functionality has been fully implemented and tested.

The CWMSPM notification might be triggered when a timer for a deadline is created or changed. This notification does not require any immediate action.

8.2.17.2 Technical description

Below is an example of the CWMSPM-notification:

```
<Notification>
        <NotificationEventType>CWMSPM</NotificationEventType>
        <NotificationSID>f94d795f-10d8-4c19-a924-b86943b28954</NotificationSID>
        <Declaration>
            <MRN>22DKRPVHOQEETGIVA8</MRN>
            <LRN>CWMSPMNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMSPMNOTIFICATION</SubmitterReferenceNumber>
            <TimerExpirationInfo>
                <ExpirationDateTime/>
            </TimerExpirationInfo>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20221107145735Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 8.2.17.1 – Notification example

Besides from the common data elements described in section 8.1.2 the following elements can be retrieved from the notification:

Element name	Description
TimerExpirationInfo	

Table 8.2.17.2 – Information in Notification

8.2.18 Error codes and warnings

When filling out the data elements for submitting a declaration, errors can happen.

If there are errors when submitting a standard declaration, **the declaration will be rejected and a CWMREJ notification will be sent with error codes indicating the error in the declaration.** The declaration should then be resubmitted by using the submission XML (the LRN can be reused in this case) with the corrected content in the data elements.

When submitting a pre-lodged declaration with errors, the declaration will not immediately be rejected – **instead the errors will be presented as warnings and sent in the CWMRCV notification.** This provides a chance to **correct the declaration before the goods are presented** (see section 6.2)**.** If the errors are not corrected before the presentation of goods, the declaration will be rejected when the goods are presented, and the declaration will have to be resubmitted as for a standard declaration.

However, a warning is not always an error. It can simply be a warning about restrictions on commodity codes or other relevant information to be aware of in the declaration.

The way to distinguish a warning code from an error code, besides looking at the type of declaration, is that warnings are given on the form **DKWxxxx**, whereas error codes are in the format **DKxxxx**, **CWMxxxxxx**, **DMSxxxxxx**, etc.

For a full list of warnings and error codes, see the document Error and Warningcodes found on github.

Error handling

9

9.1 Rejected declaration

How to handle a rejected declaration depends on the reason it was rejected. The error is described in the CWMREJ notification (see section 8.2.6), indicating which rule was broken, or which invalid data was entered in the declaration. **Standard declarations will always be instantly rejected whereas sometimes, pre-lodged declarations will receive warnings through CWMRCV notifications (see section 8.2.5).**

If the error(s) occurred from the submitter's end, the declaration can be resubmitted when the invalid data has been corrected. The LRN can be reused until the declaration is rightfully accepted (has received the CWMACC notification, see section 8.2.1).

If the error(s) occur due to system downtime or issues, the declarations can end up being rejected as well. If system downtime or service windows are not announced on **'Driftsmeddelelser'**, the main system or one of the external systems might be down, and the declaration data cannot be properly validated, resulting in rejected declarations. In this case you can contact **Toldstyrelsens Servicedesk** with information on the rejected declaration(s).

When the system is up and running again, the declarations should be resubmitted. The LRN(s) can be reused until the declaration(s) are accepted (has received the CWMACC-notification, see section 8.2.1).

9.2 Missing notifications

Sometimes the user does not receive the expected notifications. Below sections describes some common scenarios to be aware of, and how to handle them.

9.2.1 No CWMCLE notification

9.2.1.1 Specific for Import

Sometimes it happens that certain declarations do not go through to clearance (no CWMCLE notification is received) and are "stuck" in the flow after the CWMTAX notification.

The reason for that is usually that there is a manual work task for 'manual cash payment' pending for a customs officer to handle. This will occur in the following scenarios:

- **The declarant is a private person:** the EORI number in 13 05 017 000 – Declarant ID is **'DK09999981'** and 11 10 000 000 – Additional Procedure is **'C07'**.
- **The declarant is not registered for import with deferred payment:** the EORI number in 13 05 017 000 – Declarant ID is not registered for deferred payment.

In these cases, the declaration must be handled manually by a customs officer and can get clearance only when the manual cash payment task has been paid.

Unfortunately, there is currently no notification stating that the declaration has a manual cash payment pending, but there is a way to see it in the CWMTAX notification (see below).

For a **non-manual cash payment** declaration, the CWMTAX notification will look as follows:

```
<Notification>
        <NotificationEventType>CWMTAX</NotificationEventType>
        <NotificationSID>685eefec-f413-425d-a055-927856d36993</NotificationSID>
        <Declaration>
            <MRN>21DKRSYEMQS5OOTGR1</MRN>
            <LRN>CWMTAXNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMTAXNOTIFICATION</SubmitterReferenceNumber>
            <DutyTaxFee>
                <Payment>
                    <ReferenceID>DK19552101:1</ReferenceID>
                    <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                    <TaxAssessedAmount>0</TaxAssessedAmount>
                </Payment>
            </DutyTaxFee>
            <GoodsShipment>
                <GovernmentAgencyGoodsItem>
                    <SequenceNumeric>1</SequenceNumeric>
                    <Commodity>
                        <DutyTaxFee>
                            <Payment>
                                <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                                <TaxAssessedAmount currencyID="DKK">75.1</TaxAssessedAmount>
                            </Payment>
                            <SpecificTaxBaseQuantity unitCode="DKK">301</SpecificTaxBaseQuantity>
                            <DeductAmount currencyID="DKK">0</DeductAmount>
                            <TaxRateNumeric>25.0</TaxRateNumeric>
                            <TypeCode>B00</TypeCode>
                        </DutyTaxFee>
                    </Commodity>
                </GovernmentAgencyGoodsItem>
            </GoodsShipment>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210915172600Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 9.2.1.1 – Notification example

And for a **manual cash payment** declaration, the CWMTAX notification will look as follows:

```
<Notification>
        <NotificationEventType>CWMTAX</NotificationEventType>
        <NotificationSID>685eefec-f413-425d-a055-927856d36993</NotificationSID>
        <Declaration>
```

```
            <MRN>21DKRSYEMQS5OOTGR1</MRN>
            <LRN>CWMTAXNOTIFICATION</LRN>
            <VersionID>1</VersionID>
            <SubmitterReferenceNumber>CWMTAXNOTIFICATION</SubmitterReferenceNumber>
            <DutyTaxFee>
                <Payment>
                    <ReferenceID>af2c8a94-e617-11eb-a177-1eb09731c923</ReferenceID>
                    <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                    <TaxAssessedAmount>0</TaxAssessedAmount>
                </Payment>
            </DutyTaxFee>
            <GoodsShipment>
                <GovernmentAgencyGoodsItem>
                    <SequenceNumeric>1</SequenceNumeric>
                    <Commodity>
                        <DutyTaxFee>
                            <Payment>
                                <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                                <TaxAssessedAmount currencyID="DKK">75.1</TaxAssessedAmount>
                            </Payment>
                            <SpecificTaxBaseQuantity unitCode="DKK">301</SpecificTaxBaseQuantity>
                            <DeductAmount currencyID="DKK">0</DeductAmount>
                            <TaxRateNumeric>25.0</TaxRateNumeric>
                            <TypeCode>B00</TypeCode>
                        </DutyTaxFee>
                    </Commodity>
                </GovernmentAgencyGoodsItem>
            </GoodsShipment>
        </Declaration>
        <IssueDateTime>
            <DateTimeString formatCode="304">20210915172600Z</DateTimeString>
        </IssueDateTime>
    </Notification>
```

Table 9.2.1.2 – Notification example

The main difference between these two notifications can be found in the **Payment** segment on the declaration level:

Non-manual cash payment:

```
            <DutyTaxFee>
                <Payment>
                    <ReferenceID>DK19552101:1</ReferenceID>
                    <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
                    <TaxAssessedAmount>0</TaxAssessedAmount>
                </Payment>
            </DutyTaxFee>
```

Table 9.2.1.3 – Notification example

Manual cash payment:

```
<DutyTaxFee>
    <Payment>
        <ReferenceID>af2c8a94-e617-11eb-a177-1eb09731c923</ReferenceID>
        <PaymentAmount currencyID="DKK">75.1</PaymentAmount>
        <TaxAssessedAmount>0</TaxAssessedAmount>
    </Payment>
</DutyTaxFee>
```

Table 9.2.1.4 – Notification example

Notice that when there is a manual cash payment pending on the declaration, an UUID will be shown in the <ReferenceID>-element instead of the Declarant ID.

This is how one can ascertain if the reason that a declaration does not automatically go through to clearance and receives a CWMCLE notification is that it has a pending manual cash payment for a customs officer to handle.

9.2.2 No CWMTAX notification

There are certain scenarios in which a declaration does not trigger a CWMTAX notification. This can be the casewhen:

- **The submitted declaration contains an IOSS number**
- **The data element 11 10 000 000 – Additional procedure is C08 (except for tobacco, alcohol, perfume and toilet water)**

9.2.3 No CWMRCV notification/No notifications

The user should always be able to pull notifications from a given time interval. If the user keeps getting empty notifications for a correct time interval, it might be due to system downtime. If system downtime is not announced on 'Driftsmeddelelser', the main system or one of the external systems might be down, and the declaration data cannot properly be received by the system. In this case contact Toldstyrelsens Servicedesk with information on the declaration(s) with missing notifications.

When the system is up and running again, the declarations should be resubmitted. The LRN(s) can be reused until the declaration(s) are accepted (has received the CWMACC notification, see section 8.2.1).

Getting access

10

DMS can be accessed either via a system-to-system integration or by using the system's UI (called DMS Online). Regardless of which access is required, a pre-requisite is that an agreement of access has been granted by Toldstyrelsen.

For both DMS System-to-System and DMS Online you will initially get access to the test environment called TFE – Test for Erhverv. TFE can be used to prepare for production before going live on the production environment.

If you are onboarding for a system-to-system integration you will also get access to a second test environment called UFE – Udvikling for Erhverv. This environment only has system-to-system access and no access to DMS Online.

10.1 DMS Online

Login to DMS Online requires that the employee(s) who needs access has a personal employee certificate (Medarbejdercertifikat – MOCES), as login is handled through NemID. It is the LRA (Local Rights Administrator) of your company that can grant roles to the employees who need access. Roles are managed on TastSelv Erhverv here: skat.dk/tastselverhverv

On the Danish Customs and Tax Administration's GitHub we have a guide that explains how roles are assigned to the employees. The guide is in Danish and named Vejledning i brugeoprettelse til DMS.

10.2 DMS System-to-System

A system certificate (VOCES) is required to access the AS4-gateway. Our DMS Connectivity Guide found on the Danish Customs and Tax Administration's GitHub explains in detail how the connectivity to the AS4-gateway is enabled.

The system certificate needs to be granted the right roles as well. It is the LRA (Local Rights Administrator) of the economic operator that can grant roles to the employees who need access. Roles are managed on TastSelv Erhverv here: skat.dk/tastselverhverv

On the Danish Customs and Tax Administration's GitHub we have a guide that explains how roles are assigned to the employees. The guide is in Danish and named Vejledning i brugeoprettelse til DMS

10.3 User roles

When you apply for access to DMS you need to define, what type of user roles your company will need.

Overall these types of roles exist:

Role	Apply for	Description
Import Broker	DMS Import Online	Can lodge import declarations on own behalf and on behalf of others. The role is aimed at Shipping agents (Speditør).
Importer	DMS Import Online	Can lodge import declarations only on own behalf
Importer System User	DMS Import System-to-system	With this role you can lodge import declaration via DMS System-to-system

Export Broker	DMS Export Online	Can lodge export declarations on own behalf and on behalf of others. The role is aimed at Shipping agents (Speditør).
Exporter	DMS Export Online	Can lodge export declarations only on own behalf
Exporter System User	DMS Export System-to-system	With this role you can lodge Export declaration via DMS System-to-system
Transit Broker	DMS Transit Online	Can lodge transit declarations on own behalf and on behalf of others. The role is aimed at Shipping agents (Speditør).
Transit user	DMS Transit Online	Can lodge transit declarations only on own behalf
Transit System User	DMS Export System-to-system	With this role you can lodge Transit declaration via DMS System-to-system
Viewer access	DMS Online (Import, Export and Transit)	Users with this role can we any declarations that the company is a part of. Many companies that use a Shipping agent to handle their declarations has only this role.

A company can in principle possess all roles and distribute the roles individually to the employees of the company. E.g. a company has the roles “Importer”, “Exporter” and “Viewer access”. Employee “A” can then have both the “Importer” and the “Exporter roles, employee “B” may only have the “Exporter” role and employee “C” can have the “Viewer access” role.

10.3.1 Representation

If your company has one or more of the Broker roles and need to represent another party you should take special care to observe the rules laid out in the XML Guides for Import or Export. See Group 13 – Parties in the respective guide.

Verifying functionality

11

To verify the functionality of the declaration types and additional messages, as well as the ability to requests and receive notifications, we recommend that you follow the Basic Test cases found on GitHub.

11.1 XSDs and test cases

In this section you can find links to all the XSD's used for submission of declarations and for the additional messages mentioned earlier. All XSD's can be found on **GitHub**.

11.1.1 DMS Import

H7			
Type	**XSD**	**Test cases**	**XML Guide**
Submission	H7 Submission XSD	H7 Submission test cases	H7 XML Guide
Correction	H7 Correction XSD	H7 Correction test cases	H7 XML Guide
Amendment	H7 Amendment XSD	H7 Amendment test cases	H7 XML Guide
Invalidation	H7 Invalidation XSD	H7 Invalidation test cases	H7 XML Guide
Invalidation and repayment	H7 Invalidation and Repayment XSD	H7 Invalidation and Repayment test cases	In progress
Repayment and remission	Not available	Not available	Not available

Table 4.9.1 – H7 XSDs and test cases

I2			
Type	**XSD**	**Test cases**	**XML Guide**
I2 – Goods presentation	I2 - Goods Presentation XSD	I2 - Goods Presentation test cases	I2 XML Guide (for H7)

Table 4.9.2 – I2 XSDs and test cases

11.1.2 DMS Export

B1, B2, B2, B4 and C1			
Type	**XSD**	**Test cases**	**XML Guide**
Submission	Submission XSD	H7 Submission test cases	H7 XML Guide
Correction	H7 Correction XSD	H7 Correction test cases	H7 XML Guide
Amendment	H7 Amendment XSD	H7 Amendment test cases	H7 XML Guide
Invalidation	H7 Invalidation XSD	H7 Invalidation test cases	H7 XML Guide
Invalidation and repay-ment			
Repayment and remis-sion	Not available	Not available	Not available

Table 4.9.1 – Export XSDs and test cases

C2			
Type	**XSD**	**Test cases**	**XML Guide**
C2 – Goods presenta-tion			

Table 4.9.2 – C2 XSDs and test cases

Appendix

12

12.1 Flow of declarations, and related notifications

This section describes which notifications the system produces, and when in the flow the notification is produced.

The diagrams in the coming sections follow the annotation shown in figure 12.1.1 below.

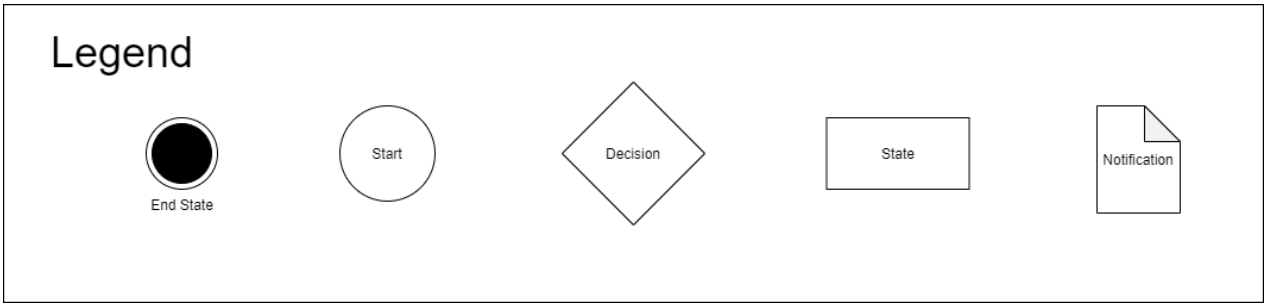

Figure 12.1.1 – Overview of diagram notation.

The End State represents a state where the flow is terminated in some respect and will not continue. An End State can trigger a notification in some cases. In most cases an End State signifies that the declaration was rejected, accepted, or converted to another flow.

The Start State signifies the starting point of a flow. There are therefore only transitions out of a Start State, and it is not possible to return, in technical terms. The start state is the first state from which it is possible for the system to issue notifications that the notification service can retrieve, meaning that it is past the semantic validation in the AS4 gateway.

A Decision State is a state from which it is possible to take one of many transitions. Only one transition will be used, and each transition out of the Decision State will be clearly labelled with the conditions necessary to take that transition. Most Decision States relate to a significant analysis performed in the flow, such as validation.

A State is the simplest construct in that it only offers one transition out, which means that this transition will always be taken.

A Notification, signified by the paper symbol, is not a state. It is therefore not possible to transition into a Notification. An arrow from any of the other symbols in the diagram to a Notification signifies that that symbol will issue a Notification which can be retrieved by the Notification service.

12.2 Import notification flows

12.2.1 H7 Notification flows

12.2.1.1 Submission

There are two kinds of declarations to submit: a standard declaration and a prelodged declaration. When the goods are presented to an IMD declaration, it will turn into an IMA declaration and go through the IMA notification flow

12.2.1.1.1 Pre-lodged IMD

The flow for IMD (pre-lodged declarations) is shown in Figure 9.2.1.1 below.

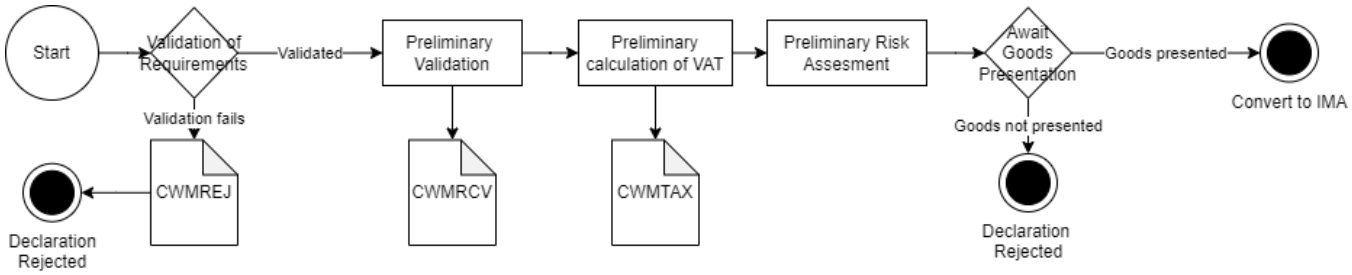

Figure 12.2.1.1 – The flow for an IMD declaration.

As shown in the diagram, an IMD declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted declaration is received.
Customs Debt Trader Notification	CWMTAX	Notification of customs debt.
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.

An IMD declaration is pre-lodged which means that information about the declaration is sent to the system before the declaration has arrived in the country. An IMD therefore also has preliminary validations, risk assessment and calculations of VAT. The IMD declaration is converted to an IMA when a goods presentation declaration has been declared. From this point the declaration therefore follows the I2 declaration flow and will therefore also produce the same notifications. To sum up, an IMD declaration will have both preliminary validation, risk assessment and calculation of VAT, and a final validation, risk assessment calculation of VAT.

12.2.1.1.2 Standard IMA

The flow for IMA (standard declarations) is shown in Figure 2 below.

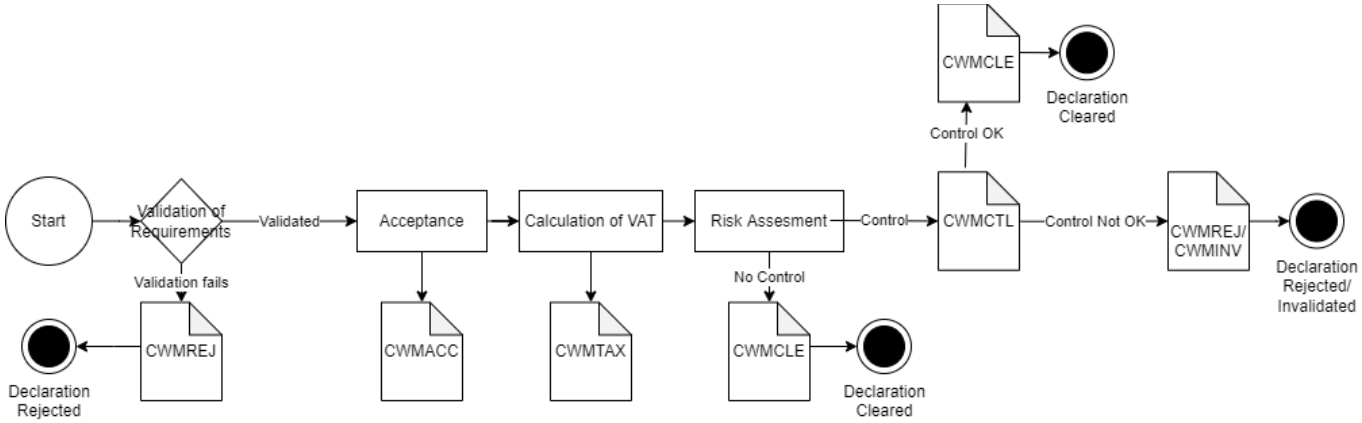

Figure 9.2.1.2 – The flow for an IMA declaration.

As shown in the diagram, an IMA declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMACC	The submitted declaration is accepted.
Customs Debt Trader Notification	CWMTAX	Notification of customs debt.
Declaration Clearance Notification	CWMCLE	Procedure is accepted and goods can be released.
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.
Declaration Control Notification	CWMCTL	Declaration has been selected for control.
Declaration Invalidation Notification	CWMINV	Declaration has been invalidated.

An IMA declaration is submitted directly as the goods are presented or following an IMD after goods are presented. The IMA has final validations, risk assessment and calculations of VAT. The declaration can be selected for control based on preliminary or final risk assessment.

12.2.1.2 Correction

Flow for Correction of a declaration is shown in Figure 9.2.1.3.

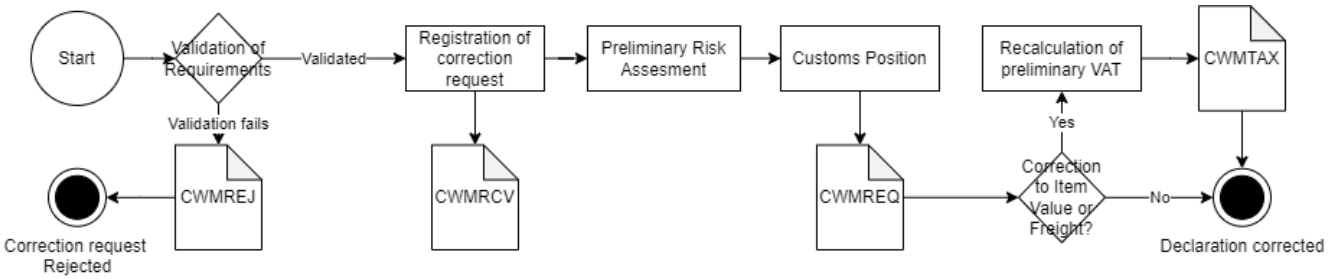

Figure 9.2.1.3 – the flow for a Correction

As shown in the diagram, a correction can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted correction application is received.
Declaration Rejection Notification	CWMREJ	Correction application has been rejected.
Customs Position on Message Notification	CWMREQ	Decision on correction application.
Customs Debt Trader Notification	CWMTAX	Notification of customs debt.

A correction can be submitted to an IMD before goods are presented. After submission of a correction the declaration will go through preliminary validations, risk assessment and, depending on if the correction was for change of item value or freight, a recalculation of VAT. If so, there will be sent a new CWMTAX notification with the preliminary VAT.

12.2.1.3 Amendment

Flow for Amendment of a declaration is shown in Figure 9.2.1.4.

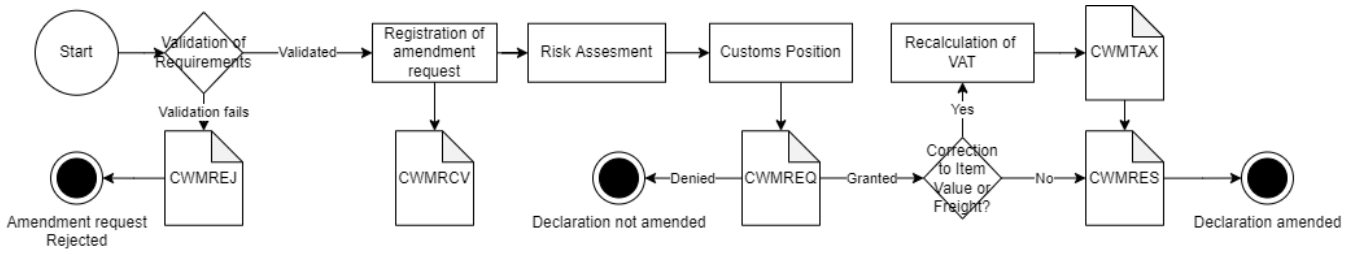

Figure 9.2.1.4 – the flow for an Amendment

As shown in the diagram, an amendment can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted amendment application is received.
Declaration Rejection Notification	CWMREJ	Amendment application has been rejected.
Customs Position on Message Notification	CWMREQ	Decision on amendment application.
Customs Debt Trader Notification	CWMTAX	Notification of customs debt.
Corrected Declaration Notification	CWMRES	The result of the amendment to the declaration.

A correction can be submitted to an IMA after goods are presented. After submission of an amendment the declaration will go through validations, risk assessment and, depending on if the amendment was for change of item value or freight, a recalculation of VAT. If so, there will be sent a new CWMTAX notification with the preliminary VAT. There will be sent a notification, CWMRES, with the results of the amendment.

12.2.1.4 Invalidation

The flow for Invalidation of a declaration is shown below in Figure 9.2.1.5.

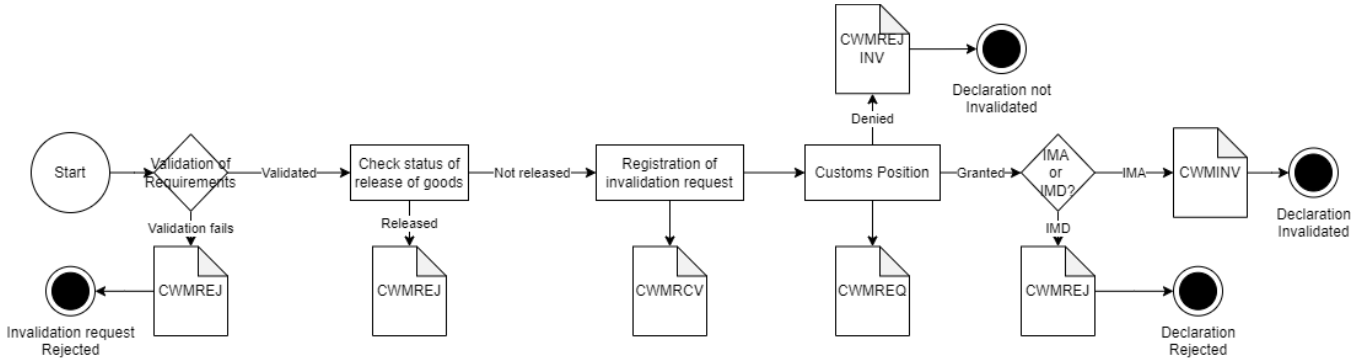

Figure 9.2.1.5 – the flow for an Invalidation

As shown in the diagram, an invalidation can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted invalidation application is received.

Declaration Rejection Notification	CWMREJ	The declaration/invalidation application has been rejected.
Customs Position on Message Notification	CWMREQ	Decision on invalidation application.
Declaration Invalidation Notification	CWMINV	Declaration has been invalidated.

The invalidation request can be sent before the release of goods. If the declaration is an IMD, the declaration will be rejected, and a new one with the same LRN can be submitted. If the declaration is an IMA the declaration will be invalidated, and a new one with the same LRN cannot be submitted, the LRN has to change.

12.2.1.5 Invalidation and Repayment

The flow for in Invalidation and Repayment of a declaration is shown below in Figure 9.2.1.6

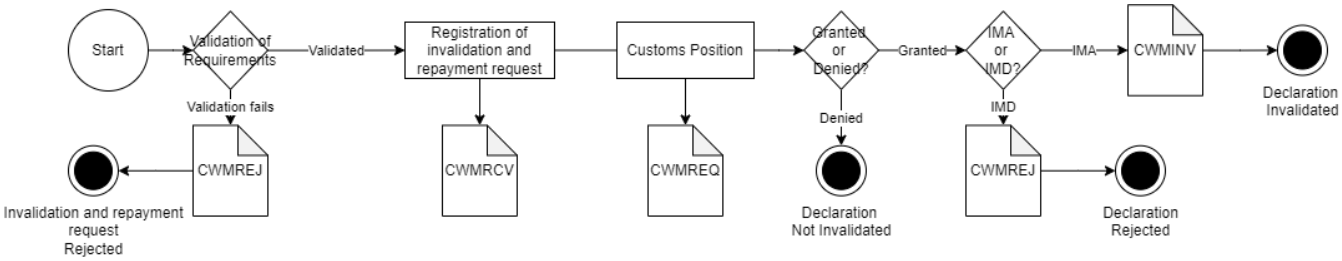

Figure 9.2.1.6 – the flow for Invalidation and Repayment

As shown in the diagram, an invalidation and repayment can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted invalidation application is received.
Declaration Rejection Notification	CWMREJ	The declaration/invalidation application has been rejected.
Customs Position on Message Notification	CWMREQ	Decision on invalidation application.
Declaration Invalidation Notification	CWMINV	Declaration has been invalidated.

The Invalidation and Repayment request can be sent after payment of VAT. If the declaration is an IMD, the declaration will be rejected, and a new one with the same LRN can be submitted. If the declaration is an IMA the declaration will be invalidated, and a new one with the same LRN cannot be submitted, the LRN has to change.

12.2.1.6 I2 – Goods presentation

The flow for an I2 Presentation Notification flow is shown below in Figure 9.2.1.7.

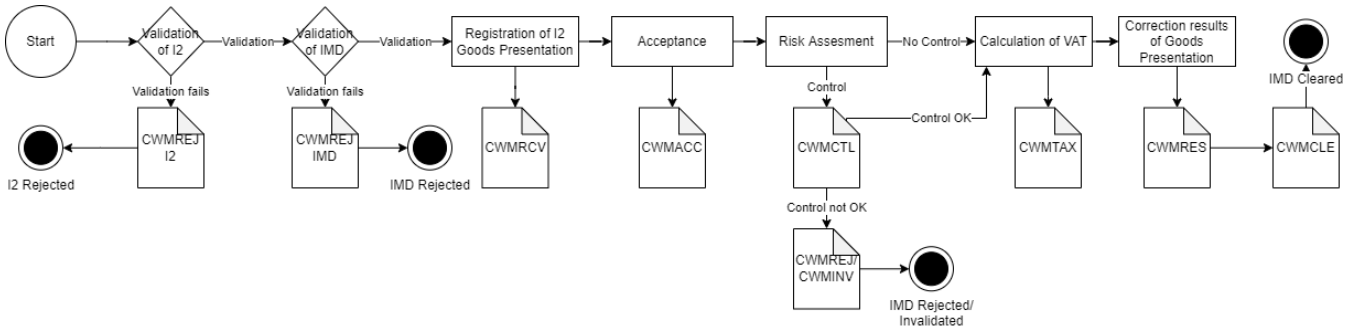

Figure 9.2.1.7 – Flow for an I2 Presentation Notification

As shown in the diagram, an I2 declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMACC	The declaration is accepted.
Customs Debt Trader Notification	CWMTAX	Notification of customs debt.
Declaration Clearance Notification	CWMCLE	Procedure is accepted and goods can be released.
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.
Declaration Accepted Notification	CWMRCV	The submitted I2 declaration is received.
Corrected Declaration Notification	CWMRES	The results of the I2 goods presentation to the declaration.
Declaration Control Notification	CWMCTL	Declaration has been selected for control.
Declaration Invalidation Notification	CWMINV	Declaration has been invalidated.

An I2 can be submitted to present goods for an IMD declaration. If the I2 declaration fails validation, only the I2 declaration is rejected, and a new I2 declaration can be submitted. If the I2 declaration passes validation, but the IMD declaration fails the final validation (i.e., as a result of not correcting non-valid data as given by warnings), the whole declaration is rejected. A new IMD/IMA declaration should be submitted if needed.
If all validations pass, the now merged declaration will go through final risk assessment and VAT calculation.
A notification with the results of the I2 goods presentation for the IMD declaration will be generated.

12.3 Export notification flows

This section is work in progress and will be updated closer to go live of the export and transit functionality

12.3.1 Notification flows

12.3.1.1 Submission

There are two kinds of declarations to submit: a standard declaration and a pre-lodged declaration. When the goods are presented to a pre-lodged declaration, it will turn into a standard declaration and go through the standard notification flow.

12.3.1.1.1 Pre-lodged

The flow for a pre-lodged declaration is shown in Figure 9.3.1.1 below.

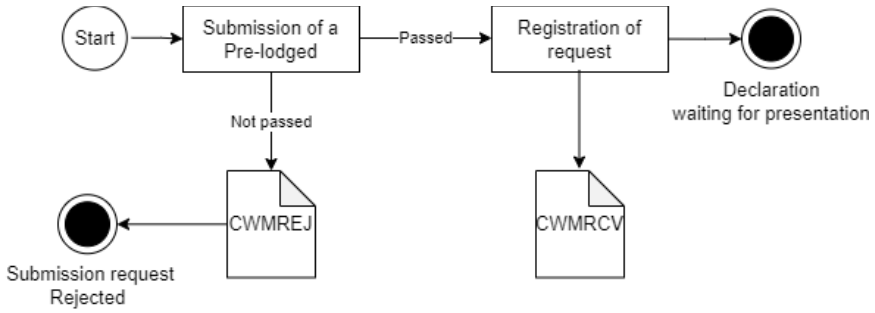

Figure 9.3.1.1 – The flow for a pre-lodged declaration

As shown in the diagram, a pre-lodged declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted declaration is received.
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.

A pre-lodged declaration means that information about the declaration is sent to the system before the declaration has left the country. The Pre-lodged declaration is converted to a standard declaration when a goods presentation declaration has been declared. If the submission is passed the request will be registered and notification CWMRCV will be sent.

12.3.1.1.2 Standard

The flow for a standard declaration is shown in Figure 9.3.1.2 below.

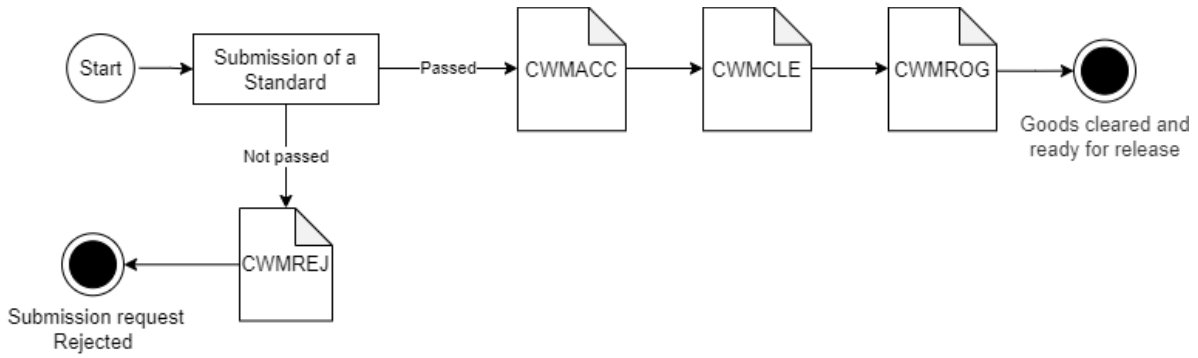

Figure 9.3.1.2 – The flow for a standard declaration

As shown in the diagram, a standard declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMACC	The declaration is accepted.
Declaration Clearance Notification	CWMCLE	Procedure is accepted and goods are cleared for release.
Declaration Release Notification	CWMROG	Notification informing the submitter that the goods are ready for released
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.

A standard declaration is submitted directly as the goods are presented or following a pre-lodged declaration after goods are presented. If the submission is passed the notifications CWMACC, CWMCLE and CWMROG will be sent.

12.3.1.2 Correction

Flow for Correction of a declaration is shown in Figure 9.2.1.3.

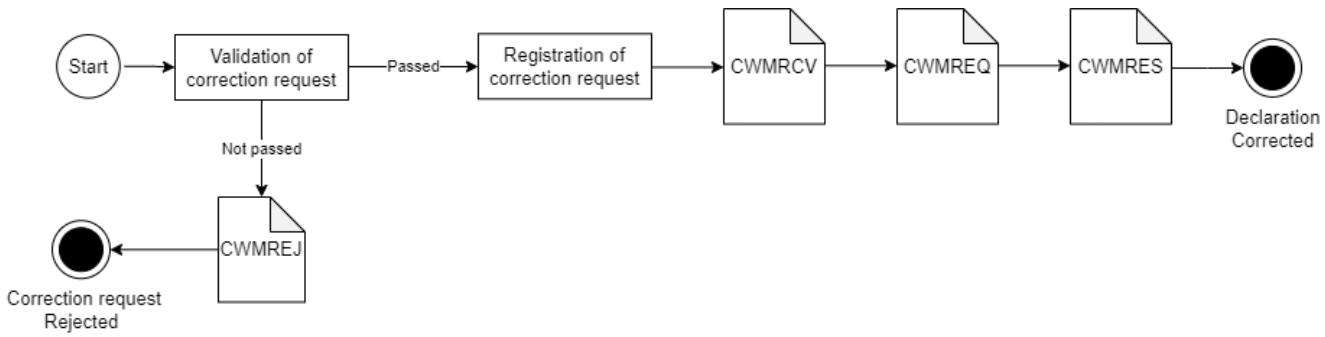

Figure 9.3.1.3 – The flow for a Correction

As shown in the diagram, a correction can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted correction application is received.
Customs Position on Message Notification	CWMREQ	Decision on correction application.

Corrected Declaration Notification	CWMRES	The results of the correction to the declaration.
Declaration Rejection Notification	CWMREJ	Correction application has been rejected.

A correction can be submitted to a pre-lodged before goods are presented. After submission of a correction the declaration will go through validation. If the validation is passed the correction will be registered and the notifications CWMRCV, CWMREQ and CWMRES will be sent.

12.3.1.3 Amendment

Flow for Amendment of a declaration is shown in Figure 9.3.1.4.

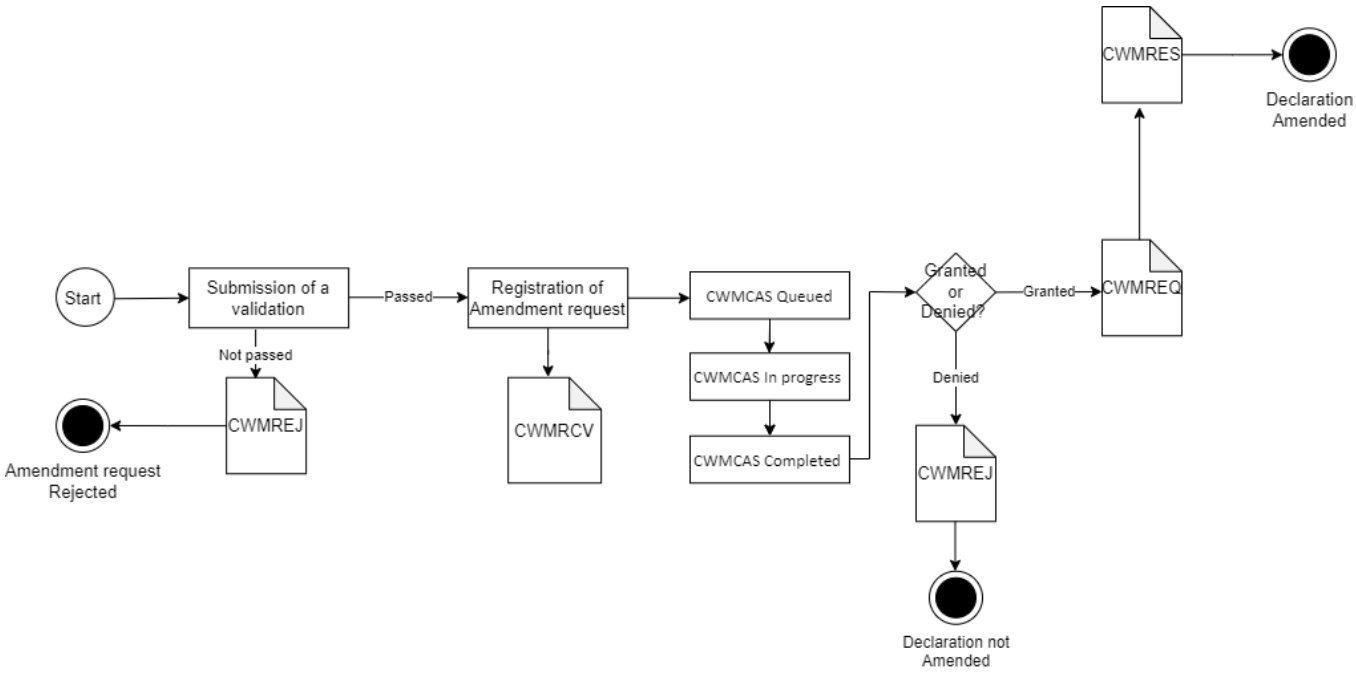

Figure 9.3.1.4 – the flow for an Amendment

As shown in the diagram, an amendment can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted amendment application is received.
Customs Position on Message Notification	CWMREQ	Decision on amendment application.
Declaration Rejection Notification	CWMREJ	Amendment application has been rejected.
Customs Position on Message Notification	CWMREQ	Decision on amendment application.
Manual Handling State Notification	CWMCAS	Notification informing the submitter about the state of a manual work task.

Corrected Declaration Notification	CWMRES	The result of the amendment to the declaration.

An Amendment can be submitted to a standard after goods are presented. After submission of an amendment the declaration will go through a validation, if the validation is passed the Amendment will be registered and the notification CWMCAS will be sent, informing about a manual work task. If the work task is granted there will be sent a CWMRES notification, meaning the declaration is amended.

12.3.1.4 Invalidation

The flow for Invalidation of a declaration is shown below in Figure 9.3.1.5.

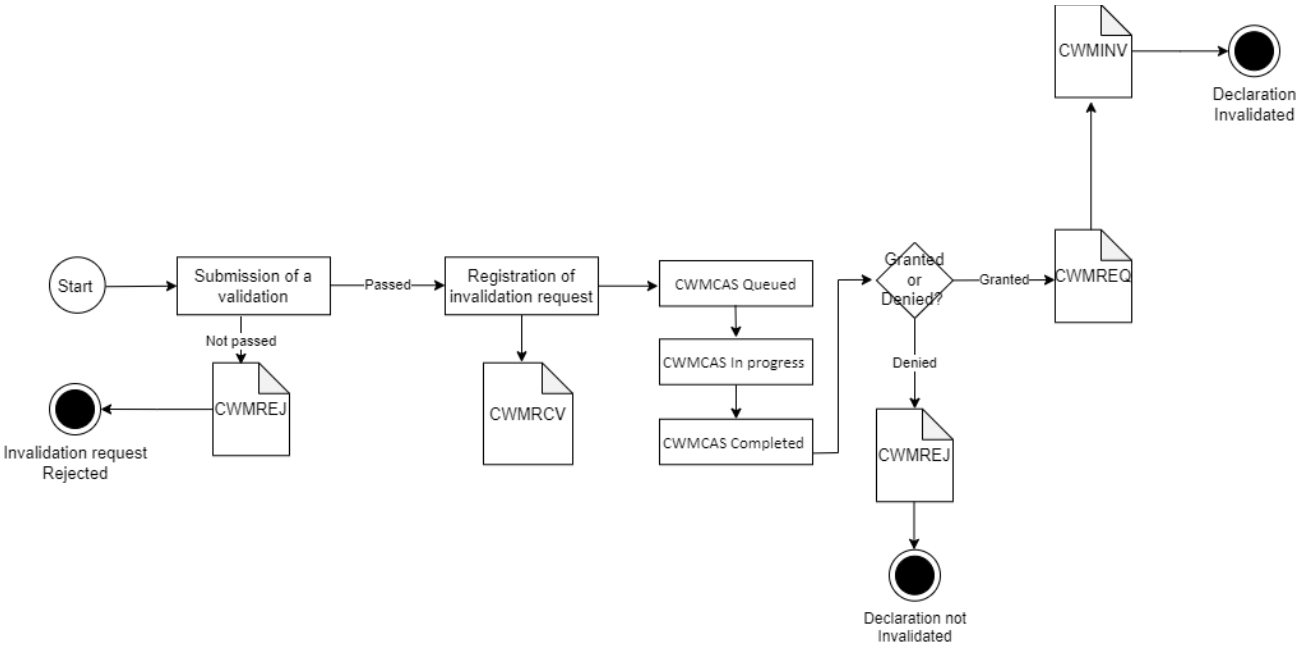

Figure 9.3.1.5 – the flow for an Invalidation

As shown in the diagram, an invalidation can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMRCV	The submitted invalidation application is received.
Declaration Rejection Notification	CWMREJ	The declaration/invalidation application has been rejected.
Manual Handling State Notification	CWMCAS	Notification informing the submitter about the state of a manual work task.
Customs Position on Message Notification	CWMREQ	Decision on invalidation application.

Declaration Invalidation Notification	CWMINV	Declaration has been invalidated.

The invalidation request can be sent before the release of goods. If the declaration is a pre-lodged, the declaration will be rejected, and a new one with the same LRN can be submitted. If the declaration is a standard, it will be invalidated, and a new one with the same LRN cannot be submitted, the LRN has to change. If the validation is passed the invalidation will be registered and the notification CWMCAS will be sent, informing about a manual work task. If the work task is granted there will be sent a CWMINV notification, meaning the declaration is invalidated.

12.3.1.5 C2 Presentation Notification

The flow for an C2 Presentation Notification flow is shown below in Figure 9.2.1.7.

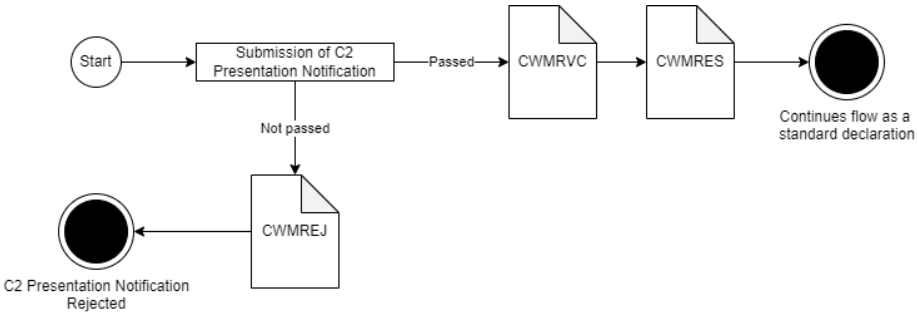

Figure 9.2.1.7 – Flow for an C2 Presentation Notification

As shown in the diagram, an C2 declaration can generate the following notifications:

Title	Code	Description
Declaration Accepted Notification	CWMACC	The declaration is accepted.
Declaration Rejection Notification	CWMREJ	Declaration has been rejected.
Declaration Accepted Notification	CWMRCV	The submitted C2 declaration is received.
Corrected Declaration Notification	CWMRES	The results of the C2 goods presentation to the declaration.

An C2 can be submitted to present goods for an pre-lodged export declaration. If the C2 declaration fails validation, only the C2 declaration is rejected, and a new C2 declaration can be submitted. If the C2 declaration passes validation, but the pre-lodged declaration fails the final validation (i.e., as a result of not correcting non-valid data as given by warnings), the whole declaration is rejected. A new declaration should be submitted if needed.

12.3.1.6 Supplementary declaration

In progress.

12.4 Exit notification flows

This section is work in progress and will be updated closer to go live of the export and transit functionality

12.5 Transit Notification flows

This section is work in progress and will be updated closer to go live of the export and transit functionality

12.6 AS4 Services

The following section describes the available services provided by the AS4-gateway. The parameters MessageProperties[procedureType] and Service.Action in the AS4 header allows setting of which service the AS4 message is destined for.

Each section describes the services to be used for the various system environments we use.

- Production environment
- TFE (Test For Erhverv) test environment
- UFE (Udvikling For Erhverv) test environment

12.6.1 Services for production environment

12.6.1.1 Export services and actions

Function	Service	Action	Types of declaration
Submit export declaration	DMS.Export	Declaration.Submit	B1, B2, B3, B4, C1
Amend/Correct export declaration	DMS.Export	Declaration.Amend	B1, B2, B3, B4, C1
Invalidate export declaration	DMS.Export	Declaration.Invalidate	B1, B2, B3, B4, C1
Goods Presentation notification for export declaration	DMS.Export	Declaration.Amend.Goodspresented	B1, B2, B3, B4, C1
Supplement declaration for export declaration	DMS.Export	Declaration.Amend.Supplement	B1, B2, B3, B4, C1
Retrieve notifications	DMS.Export	notification	All
Submit exit declaration	DMS.Exit	Declaration.Submit	A1, A2, A3
Amend/Correct exit declaration	DMS.Exit	Declaration.Amend	A1, A2, A3
Invalidate exit declaration	DMS.Exit	Declaration.Invalidate	A1, A2, A3
Arrival Notification for exit declaration	DMS.Exit	Declaration.Arrival	A1, A2, A3
Exit Notification	DMS.Exit	Declaration.Notification	A1, A2, A3

12.6.1.2 Transit services and actions

Function	service	Action	Types of declaration
Submit transit declaration	DMS.Transit	Declaration.Submit	D1, D2, D3, D4

Amend/Correct transit declaration	DMS.transit	Declaration.Amend	D1, D2, D3, D4
Invalidate transit declaration	DMS.transit	Declaration.Invalidate	D1, D2, D3, D4
Goods Presentation notification for transit declaration	DMS.transit	Declaration.Goodspresented	D1, D2, D3, D4
Retrieve notifications	DMS.transit	notification	All

12.6.1.3 Import services and actions

Function	service	Action	Types of declaration
A create declaration	DMS.Import2	Declaration.Submit	H7
An amendment for a declaration	DMS.Import2	Declaration.Amend	H7
Goods Presentation notification for import declaration	DMS.Import2	Declaration.Amend.Goodspresented	I2
Invalidation message	DMS.Import2	Declaration.Invalidate	H7
Invalidation and repayment message	DMS.Import2	Declaration. InvalidateRemissionRepayment	H7
Retrieve the latest notifications.	DMS.Import2	Notification	All

12.6.2 Services for TFE environment

12.6.2.1 Export services and actions

Function	service	Action	Types of declaration
Submit export declaration	DMS.Export	Declaration.Submit	B1, B2, B3, B4, C1
Amend/Correct export declaration	DMS.Export	Declaration.Amend	B1, B2, B3, B4, C1
Invalidate export declaration	DMS.Export	Declaration.Invalidate	B1, B2, B3, B4, C1
Goods Presentation notification for export declaration	DMS.Export	Declaration.Amend.Goodspresented	B1, B2, B3, B4, C1
Supplement declaration for export declaration	DMS.Export	Declaration.Amend.Supplement	B1, B2, B3, B4, C1

Retrieve the latest notifications	DMS.Export	notification	All
Submit exit declaration	DMS.Exit	Declaration.Submit	A1, A2, A3
Amend/Correct exit declaration	DMS.Exit	Declaration.Amend	A1, A2, A3
Invalidate exit declaration	DMS.Exit	Declaration.Invalidate	A1, A2, A3
Arrival Notification for exit declaration	DMS.Exit	Declaration.Arrival	A1, A2, A3
Exit Notification	DMS.Exit	Declaration.Notification	A1, A2, A3

12.6.2.2 Transit services and actions

Function	service	Action	Types of declaration
Submit transit declaration	DMS.Transit	Declaration.Submit	D1, D2, D3, D4
Amend/Correct transit declaration	DMS.transit	Declaration.Amend	D1, D2, D3, D4
Invalidate transit declaration	DMS.transit	Declaration.Invalidate	D1, D2, D3, D4
Goods Presentation notification for transit declaration	DMS.transit	Declaration.Goodspresented	D1, D2, D3, D4
Retrieve the latest notifications	DMS.transit	notification	All

12.6.2.3 Import services and actions

Function	service	Action	Types of declaration
Submit import declaration	DMS.Import2	Declaration.Submit	H7
Amend/Correct import declaration	DMS.Import2	Declaration.Amend	
Goods Presentation notification for import declaration	DMS.Import2	Declaration.Amend.Goodspresented	I2
Invalidate import declaration	DMS.Import2	Declaration.Invalidate	
Invalidation and repayment for import declaration	DMS.Import2	Declaration. InvalidateRemissionRepayment	
Retrieve notifications.	DMS.Import2	Notification	

12.6.3 Services for UFE environment

12.6.3.1 Export services and actions

Function	Service	Action	Types of declaration
Submit export declaration	DMS.Export2	Declaration.Submit	B1, B2, B3, B4, C1
Amend/Correct export declaration	DMS.Export2	Declaration.Amend	B1, B2, B3, B4, C1[1]
Invalidate export declaration	DMS.Export2	Declaration.Invalidate	B1, B2, B3, B4, C1[1]
Goods Presentation notification for export declaration	DMS.Export2	Declaration.Amend.Goodspresented	B1, B2, B3, B4, C1 [1]
Supplement declaration for export declaration	DMS.Export2	Declaration.Amend.Supplement	C1[1]
Retrieve notifications	DMS.Export2	Notification	All
Submit exit declaration	DMS.Exit2	Declaration.Submit	A1, A2, A3
Amend/Correct exit declaration	DMS.Exit2	Declaration.Amend	A1, A2, A3[1]
Invalidate exit declaration	DMS.Exit2	Declaration.Invalidate	A1, A2, A3[1]
Arrival Notification for exit declaration	DMS.Exit2	Declaration.Arrival	A1, A2, A3[1]
Exit Notification	DMS.Exit2	Declaration.Notification	A1, A2, A3[1]

12.6.3.2 Transit services and actions

Function	service	Action	Types of declaration
Submit transit declaration	DMS.Transit2	Declaration.Submit	D1, D2, D3, D4
Amend/Correct transit declaration	DMS.transit2	Declaration.Amend	D1, D2, D3, D4

[1] For the additional messages (.Amend, .Invalidate, .Amend.Goodspresented, and .Amend.Supplement), several declaration types are written. This is because the endpoints in question apply to all of these declaration types. For example, the .Invalidate endpoint can be used to invalidate B1, B2, B3, B4, and C1 declarations.

Invalidate transit declaration	DMS.transit2	Declaration.Invalidate	D1, D2, D3, D4
Goods Presentation notification for transit declaration	DMS.transit2	Declaration.Goodspresented	D1, D2, D3, D4
Retrieve notifications	DMS.transit2	notification	All

12.6.3.3 Import services and actions

Import services and actions not available for UFE.

UDVIKLINGS OG FORENKLINGS STYRELSEN

www.ufst.dk
ISBN [000-00-0000-0000-0]

Udviklings- og Forenklings-styrelsen er en del af Skatteforvaltningen