|  |
| --- |
| Connectivity Guide |
| DMS H7 |

|  |
| --- |
| Test Scenarios January 2021 |
|  |

# Document version

|  |
| --- |
|  |
| Document version:   |  |  |  |  | | --- | --- | --- | --- | | Version nr. | Dato | Created by | Updates | | 1.0 | 25/01/21 | Bjarke Runz Jensen and Christian Bach Møllnitz |  | | 1.1 | 08-02-2021 | Bjarke Runz Jensen | Sec. 5.1.1. Temporary protocol change  Sec. 7.3 Certificate UI change | | 1.2 | 23-02-2021 | Christian Bach Møllnitz | Added section 6.4 with submitter info.  Added section 7 with AS4 response details.  Clarified section 6.1 with payload location.  Clarified section 8.1 on encryption.  Added section 8.5.2 for correct JKS keystore creation.  Removed section 8.5.3. | | 1.3 | 26-02-2021 | Bjarke Runz Jensen | The connection is now encrypted | | 1.4 | 04-03-2021 | Christian Bach Møllnitz | Added Notification Chapter 7  Added Notification push pmode Section 10.2 | | 1.5 | 22-03-2021 | Christian Bach Møllnitz | Updated AS4 Header and XML payload examples | | 1.6 | 21-04-2021 | Christian Bach Møllnitz | Added synchronous AS4 examples in section 11.1  Expanded chapter 8 with StatusResponse XSD  Added Notification response demo. | | 1.7 | 10-05-2021 | Christian Bach Møllnitz | Added prod URL  Updated notification examples | |

Table of Content

[1 Document version 1](#_Toc69902612)

[2 Purpose 4](#_Toc69902613)

[3 Overview 4](#_Toc69902614)

[4 Connectivity Test Cases 4](#_Toc69902615)

[5 Server Details 5](#_Toc69902616)

[5.1 AS4 End Point 5](#_Toc69902617)

[5.1.1 VOCES CVR Users: 5](#_Toc69902618)

[5.1.2 EORI Users 5](#_Toc69902619)

[5.2 Testing Connectivity 5](#_Toc69902620)

[5.2.1 Unix 5](#_Toc69902621)

[5.2.2 Windows 6](#_Toc69902622)

[6 Delivery Method 11](#_Toc69902623)

[6.1 AS4 Message 11](#_Toc69902624)

[6.2 AS4 Header 12](#_Toc69902625)

[6.3 AS4 Services 15](#_Toc69902626)

[6.4 Submitter 15](#_Toc69902627)

[7 Notifications 15](#_Toc69902628)

[8 Response Method 17](#_Toc69902629)

[9 Security 17](#_Toc69902630)

[9.1 Signing 17](#_Toc69902631)

[9.2 AS4 Username and Password 18](#_Toc69902632)

[9.3 Registering Client Certificate 18](#_Toc69902633)

[9.4 Certificate Renewal 21](#_Toc69902634)

[9.5 Certificates 22](#_Toc69902635)

[9.5.1 User Certificate 22](#_Toc69902636)

[9.5.2 Correct conversion from .p12 to .jks certificate 22](#_Toc69902637)

[10 P-Modes 23](#_Toc69902638)

[10.1 Declaration Push P-mode 24](#_Toc69902639)

[10.2 Notification Push P-mode 26](#_Toc69902640)

[11 XML Examples 27](#_Toc69902641)

[11.1 Synchronous answers example: 27](#_Toc69902642)

[11.1.1 Unapproved message: 27](#_Toc69902643)

[11.1.2 Approved message: 27](#_Toc69902644)

[12 Possible Errors 27](#_Toc69902645)

[12.1 DMS Fails to Authenticate User 27](#_Toc69902646)

[12.2 Certificate Errors 28](#_Toc69902647)

# Purpose

This document describes the details for delivering files to the DMS-Import system. The document is described at a time, where H7 only functionality is implemented, and as such, examples and delivery methods are directed towards that, but built in a way that should allow for other declarations types in the future.

The document describes server details, and tests steps to confirm a working connection towards the system. The system accepts messages following the AS4 standard. This document describes the general aspects of AS4, the needed AS4-header, security, and attachment setup. Lastly follows common errors, and their resolution.

# Overview

DMS Import is accessed through SKAT’s common B2Bi solution, via its established AS4 endpoints. AS4 is a general format and can be used to transfer any payload in a secure manner. AS4 standardizes exchanges in a synchronous and asynchronous setup, and the DMS B2B solution utilizes both.

Transferring messages to the B2Bi system requires a normal one-way HTTPS connection, as well as encrypting and signing the AS4 massages using the [WS-\*](https://en.wikipedia.org/wiki/List_of_web_service_specifications) standard AS4 dictates.

# Connectivity Test Cases

This section describes specific testcases to do, in order to have a fully passed connectivity test. In order to move successfully to the next test-phase, the described test-cases must be passed successfully.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test number** | **Test description** | **Section** | **Passed** |
| **1** | Acquire VOCES certificate | 7.5.1 User Certificate | ☐ |
| **2** | Acquire B2B password | 7.2 AS4 Username and password | ☐ |
| **3** | Acquire server certificate | 7.5.2 Encryption Certificate | ☐ |
| **4** | Register Client Certificate | 7.3 Registering Client Certificate | ☐ |
| **5** | Test network access | 5.2 Testing connectivity | ☐ |
| **6** | Verify AS4 header filled correctly | 6.2 AS4 header | ☐ |
| **7** | Verify AS4 header signing | 7.1 Signing and encryption | ☐ |
| **8** | Verify AS4 header encryption | 7.1 Signing and encryption | ☐ |

# Server Details

|  |  |  |
| --- | --- | --- |
| Hostname | Port | IP |
| secureftpgatewaytest.skat.dk | 6384 | 195.85.251.58 |
| secureftpgateway.skat.dk | 6384 | 195.85.251.102 |

The client needs access to this server, on the correct port. As far as possible, the client needs to resolve the hostname on suitable nameserver. The IP listed above is the currently active IP at the time of writing. The given IP is suitable for change.

## AS4 End Point

The endpoint used depends on the CVR/EORI and UID of the client certificate, the format is described below.

### VOCES CVR Users:

https://<hostname>:6384/exchange/CVR\_<CVR>\_UID\_<UID>

Example:

<https://secureftpgateway.skat.dk:6384/exchange/CVR_30808460_UID_25351738>

### EORI Users

https://<hostname>:6384/exchange/EORI\_<EORI>\_RID\_<RID>

Example:

<https://secureftpgateway.skat.dk:6384/exchange/EORI_SE4445462718_RID_1391404656315>

## Testing Connectivity

The following section describes various tools and methods for verifying connectivity from the client towards the DMS B2Bi solution. Availability of tools on the client setup will determine the method for testing.

### Unix

This section describes ways to test the connectivity on Unix-style servers, using common connectivity testing tools.

Method #1 – telnet

telnet <Hostname> 6384

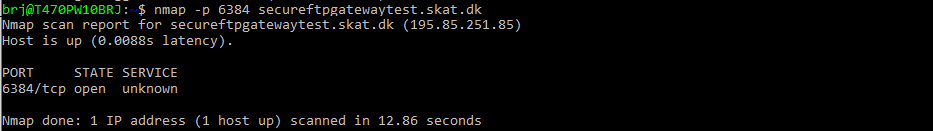
Example:



Method #2 – nmap

nmap -p 6384 <Hostname>

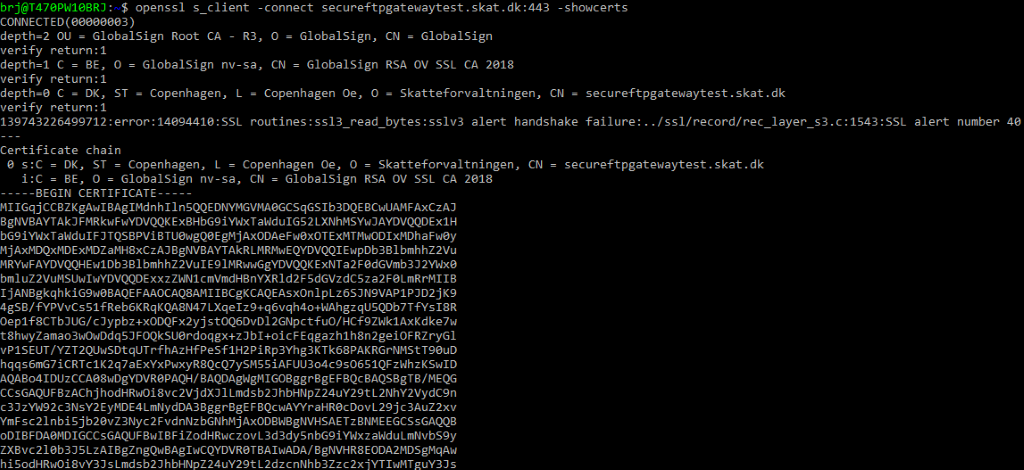
Example:



Method #3 – openssl

openssl s\_client -connect <Hostname>:443 -showcerts

Example:

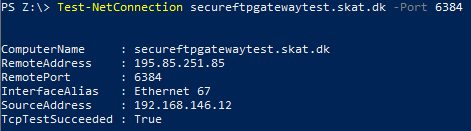


### Windows

This section describes ways to test the connectivity on Windows-style servers, using common connectivity testing tools.

Method #1 - Test-NetConnection [Requires execution in Powershell]

Test-NetConnection <Hostname>-Port 6384

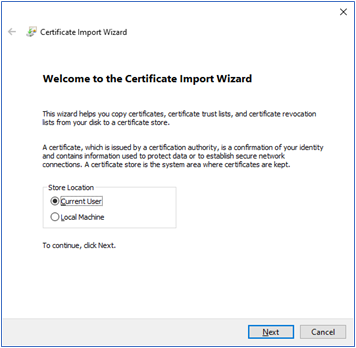


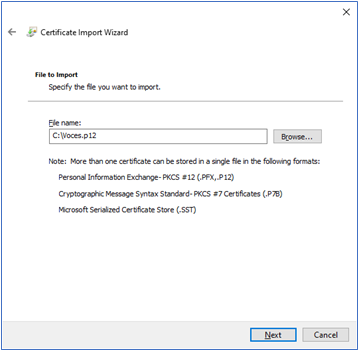
Method #2 – Browser

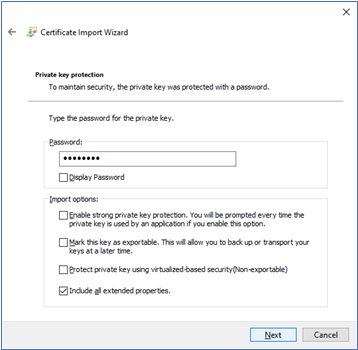
Requires valid VOCES (or similar trusted client certificate in testing clients truststore).

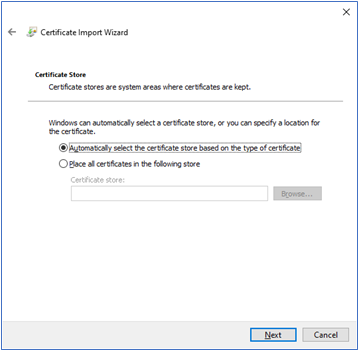
Installation of client certificate on client PC:

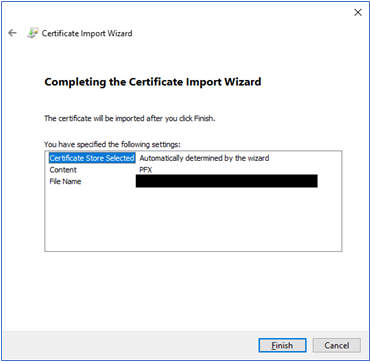
1. Locate suitable certificate. See details section 7.5.
2. Install certificate in truststore – if access allows, double-click the pfx/p12 file, and see this menu:

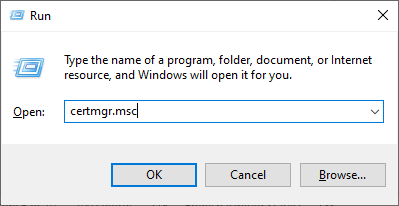
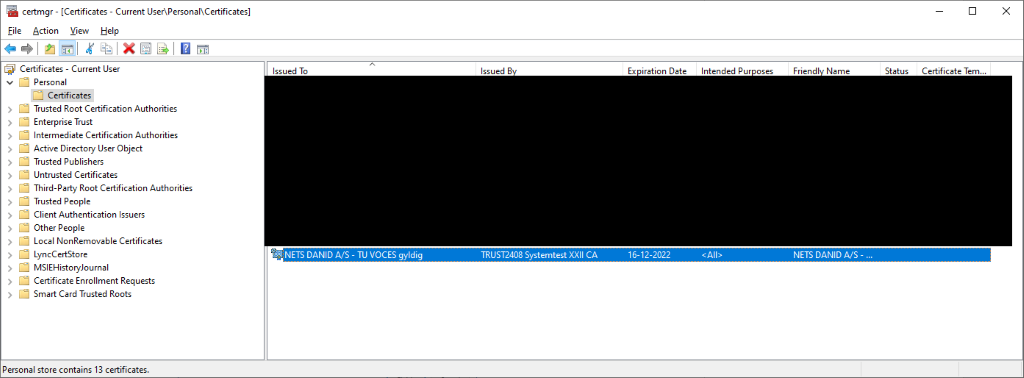
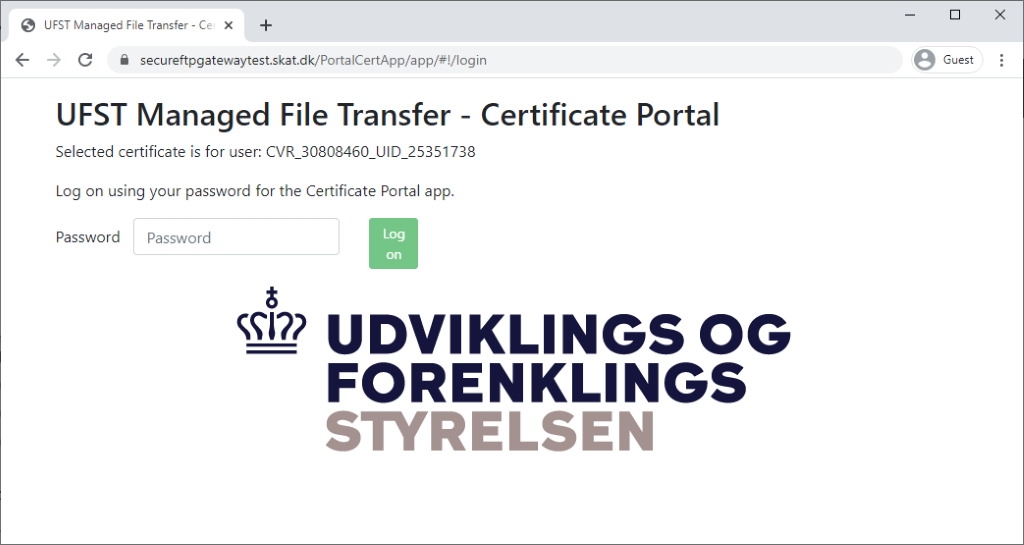










1. Verify installation in Widows Certificate manager:
2. Open Certificate Manager:   
   
3. Locate Certificate. Certificate Store and naming of client Certificate depends on client setup:
4. Open <https://secureftpgateway.skat.dk> in favourite browser, that has access to the internet, on a client setup that the internal network is setup as the accessing system:   
   

# Delivery Method

The following section describes how AS4 works, and details needed configuration, in order to access the system.

## AS4 Message

AS4 is a standard describing various fields related to the message transfer – described in a header.

AS4 furthermore standardizes encryption and signing of the payload, using the WS-\* standard. AS4 is closely related to SOAP, in that it utilizes the soap-envelope for defining headers and payload elements. The main difference from AS4 has from soap, is that **there is no soap-WSDL** describing the service. This means that there is not a single file to help define the complete service schemas and endpoints. These settings require manual setup, to define the following:

* AS4 header XSD
* Payload XSD (either declaration or notification)
* Endpoint
* Encryption settings

The setup of the following depends wholly on the client setup – there exist many implementations of AS4-clients, and how these settings are applied, is determined by the client.

A list of existing open source AS4-clients can be found on:

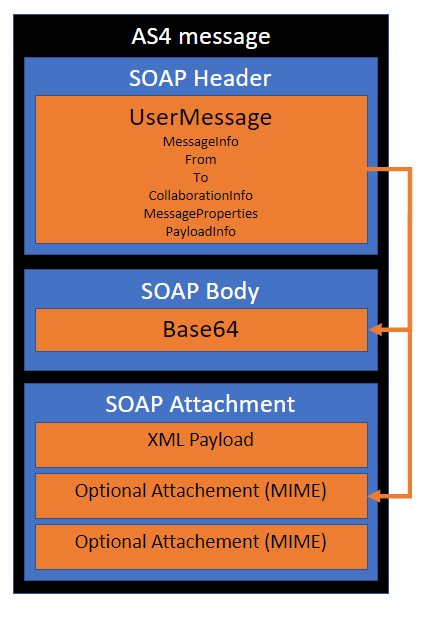
<https://peppol.eu/downloads/peppolimplementations/>

and on:

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4+conformant+solutions>

The Holodeck AS4 solution has been used internally at UFST for testing the AS4 delivery-method.

An AS4 message allows sending of a payload **only** as an attachment to the message, and not in the body. In DMS the main payload – the declaration, or request for notification is set to be delivered in the soap-body. See XML samples in section 11 XML examples. The message structure is shown here below:



The soap attachments are planned for usage for amendments in a later release – and will not be tested as part of the testcases described here.

## AS4 Header

The AS4 header XSD, can be downloaded from this URL: <https://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms-header-3_0-200704.xsd>

Some clients come with this header preloaded.

This section contains information about what information should be contained in the AS4 header.

The following attributes must be provided, the **bolded values must not be changed,**the rest depends on the client certificate and user:

|  |  |  |
| --- | --- | --- |
| Attribute | Value | Example |
| MessageInfo.Timestamp | YYYY-MM-DDTHH:MI:ss.SSSZ | 2021-01-19T15:24:37.376Z |
| MessageInfo.MessageId | GUID@CVR\_<CVR>\_UID\_<UID> | d4872030-3862-4e7b-9754-17a98523e826@CVR\_30808460\_UID\_25351738 |
| PartyInfo.From.PartyId | CVR\_<CVR>\_UID\_<UID>**\_AS4** | CVR\_30808460\_UID\_25351738**\_AS4** |
| PartyInfo.From.Role | **AS4 Initator Role** | **http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator** |
| PartyInfo.To.PartyId | **AS4 receiver** | **SKAT-MFT-AS4** |
| PartyInfo.To.Role | **AS4 Initator Role** | **http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder** |
| CollaborationInfo. Service | Service prefix | DMS.Import (se AS4 service section for details) |
| CollaborationInfo. Action | Service Postfix (Action) | Declaration.Submit (se AS4 service section for details) |
| CollaborationInfo. ConversationId | GUID | f411f3b5-26ff-4207-baf9-a50526d9063f |
| MessageProperties. Property[lang] | **Language** | **EN** |
| MessageProperties. Property[procedureType] | ProcedureType | H7 (se AS4 service section for details) |
| PayloadInfo.PartInfo. PartProperties.Property[original-file-name] | File name | im\_decl\_01.01.2021\_0001.xml |

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

<eb3:Messaging xmlns:mustUnderstand=**"http://www.w3.org/2003/05/soap-envelope"** xmlns:eb3=**"http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"** xmlns:wsu=**"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"** mustUnderstand:mustUnderstand=**"true"** wsu:Id=**"id-4b2850335f374e5-f471-4f64-9e3d-e1b845277dd9"**>

<eb3:UserMessage>

<eb3:MessageInfo>

<eb3:Timestamp>**2021-01-19T15:24:37.376Z**</eb3:Timestamp>

<eb3:MessageId>**d4872030-3862-4e7b-9754-17a98523e826@CVR\_30808460\_UID\_25351738**</eb3:MessageId>

</eb3:MessageInfo>

<eb3:PartyInfo>

<eb3:From>

<eb3:PartyId type=**"string"**>**CVR\_30808460\_UID\_25351738\_AS4**</eb3:PartyId>

<eb3:Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator**</eb3:Role>

</eb3:From>

<eb3:To>

<eb3:PartyId type=**"string"**>**SKAT-MFT-AS4**</eb3:PartyId>

<eb3:Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder**</eb3:Role>

</eb3:To>

</eb3:PartyInfo>

<eb3:CollaborationInfo>

<eb3:Service type=**"string"**>**DMS.Import**</eb3:Service>

<eb3:Action>**Declaration.Submit**</eb3:Action>

<eb3:ConversationId>**f411f3b5-26ff-4207-baf9-a50526d9063f**</eb3:ConversationId>

</eb3:CollaborationInfo>

<eb3:MessageProperties>

<eb3:Property name=**"lang"**>**EN**</eb3:Property>

<eb3:Property name=**"procedureType"**>**H7**</eb3:Property>

</eb3:MessageProperties>

<eb3:PayloadInfo>

<eb3:PartInfo>

<eb3:PartProperties>

<eb3:Property name=**"original-file-name"**>**im\_decl\_01.01.2021\_0001.xml**</eb3:Property>

</eb3:PartProperties>

</eb3:PartInfo>

</eb3:PayloadInfo>

</eb3:UserMessage>

</eb3:Messaging>

## AS4 Services

The following section describes the available services provided by AS4. The parameters MessageProperties[procedureType] and Service.Action in the AS4 header allows setting of which service the AS4 message is destined for. The parameters given ensures correct and immediate payload XML Schema validation, as well as ensuring the correct internal flow for processing the file is started upon receival.

|  |  |
| --- | --- |
| BusinessService | Internal schema and processing  engine |
| DMS.Import.Declaration.Submit | A create declaration (for now only H7) |
| DMS.Import.Declaration.Amend | An amendment for a declaration |
| DMS.Import.Declaration.Amend.Goodspresented | I2 declaration message (Only for selected clients) |
| DMS.Import.Declaration.Invalidate | Invalidation message |
| DMS.Import.Declaration.InvalidateRemissionRepayment | Invalidation and repayment message |
| DMS.Import.Notification | Retrieve the latest notifications. |

## Submitter

In every call to Axway a submitter needs to be filled out in the payload, or in the AS4 header, for all companies the submitter name will be the numbers in their CVR / EORI number, for example, if a company has the following CVR: CVR\_**30808460** then the submitter name will be **30808460**, and will look like this in the payload:

<ns2:Submitter>

<ns2:Name>**30808460**</ns2:Name>

</ns2:Submitter>

# Notifications

With the service **DMS.Import.Notification** notifications can be retrieved from Axway.

In order to retrieve notifications, a declaration has to be submitted with the service **DMS.Import.Declaration.Submit**.

The **DMS.Import.Notification** service has the following properties for a valid request:

* **functionalReferenceId** – Otherwise referred to as LRN, **only for debugging**!
* **submitterId** – the identification of the submitter.
* **dateFrom** – the date from which notifications should be retrieved in the format **yyyy/mm/dd/THH:MM:SS.MSS** for example: 2021-04-16T15:45:00.000
* **dateTo** – the date to from where notifications should be retrieved in the format **yyyy/mm/dd/THH:MM:SS.MSS** for example: 2021-04-16T15:45:00.000

Below you will find an example of an XML payload, sent to Axway, which retrieves all available notifications for the declarations which were sent in by the trader “**30808460**” between the timestamps **2021-05-07T06:00:00.000** and **2021-05-07T13:00:00.000.**

Additionally, included is the SIT01TimeNotifications file, which is a the answer provided from Axway, as a response to the request.



Double click to open file.

# Response Method

When a user message is sent from Axway to the operator, a service and action will need to be provided in the AS4 header as follows:

* Synchronous answers, for Submit, Amend, and error-messages in Notifications etc.
  + Service: [http://docs.oasis-open.org/ebxml-msg/as4/200902/service](https://eur02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fdocs.oasis-open.org%2Febxml-msg%2Fas4%2F200902%2Fservice&data=04%7C01%7Ccbm%40netcompany.com%7Cba8a500abac74540255b08d8d7da1ba3%7C8f9b88a73f3e4be3aae42006d4c42306%7C1%7C0%7C637496679071958425%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2FkdTRoHjmY4UKuwh%2FkZpdOt%2BVEPxuiGnOT%2B4U8unp1w%3D&reserved=0)
  + Action: statusResponse
* Answers for fetching of Notifications
  + Service: [http://docs.oasis-open.org/ebxml-msg/as4/200902/service](https://eur02.safelinks.protection.outlook.com/?url=http%3A%2F%2Fdocs.oasis-open.org%2Febxml-msg%2Fas4%2F200902%2Fservice&data=04%7C01%7Ccbm%40netcompany.com%7Cba8a500abac74540255b08d8d7da1ba3%7C8f9b88a73f3e4be3aae42006d4c42306%7C1%7C0%7C637496679071958425%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=%2FkdTRoHjmY4UKuwh%2FkZpdOt%2BVEPxuiGnOT%2B4U8unp1w%3D&reserved=0)
  + Action: Response

Each synchronous answer consists of a matched set of two messages that arrive at roughly the same time.

The ‘pi’ message contains the actual status of the sent message, contained in a StatusResponse and Code, while the ‘mi’ message contains timestamp and messageID.

Here is the XSD schema for the StatusResponse element:



An XML example of synchronous answers can be found in section 11.1

# Security

## Signing

The following webpage describes detailed the security aspects about AS4: <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4+-+1.14#eDeliveryAS4-1.14-Security>

In general, the DMS solution expects the following elements being signed:

* Body
* Messaging
* cid:Attachments

The solution has been tested using hash-function/digest-method: xmlenc#sha256 – and signature Algorithm: xmldsig-more#rsa-sha256.

The solution does **not**(!) use encryption for the xml messages.

See example messages in Section 9. XML examples.

## AS4 Username and Password

The username and password is obtained through the selfservice portal as part of the certificate registration process. See section below.

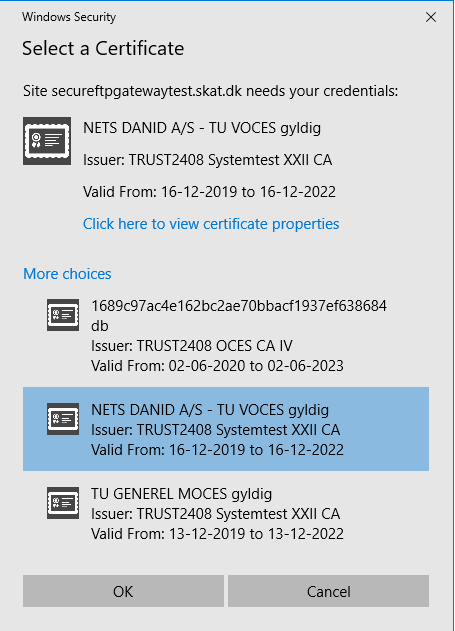
## Registering Client Certificate

The Certificate Portal provides self-service for pre-registration of certificates.

TFE: [https://secureftpgatewaytest.skat.dk](https://secureftpgatewaytest.skat.dk/)

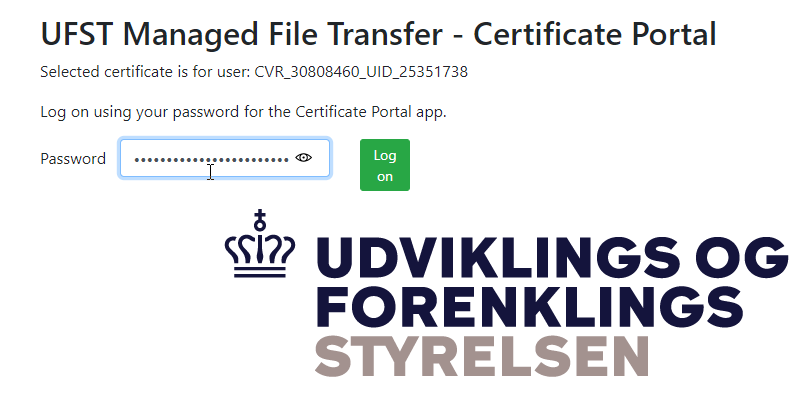
PROD: [https://secureftpgateway.skat.dk](https://secureftpgateway.skat.dk/)

You are required to use the same certificate as you would use for the FTPS Gateway.

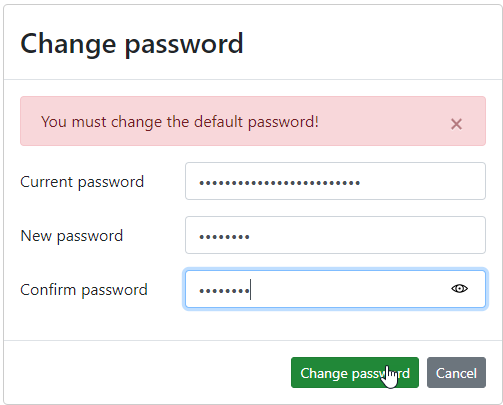


In above example (from IE) you can select from the certificates, which have been imported to the browser. Here we select a NETS test certificate and enter the logon page of the Certificate Portal. The CVR and UID/RID information is extracted from the certificate and you are identified as user: CVR\_30808460\_UID\_25351738.

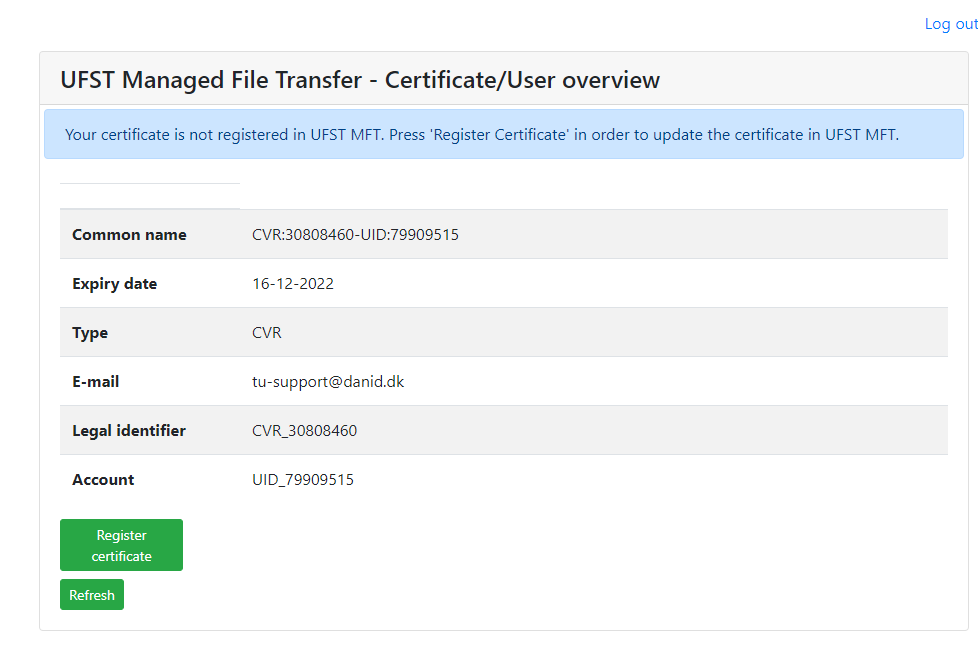
The first time you logon the default password is your user identification, and you may thus simply copy/paste and proceed with logon.



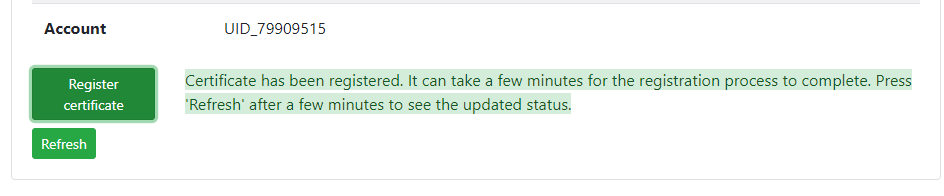
The first time you login you are requested to change password. You may use the passphrase for your certificate or any other password, which will thus be required for subsequently   
logon.



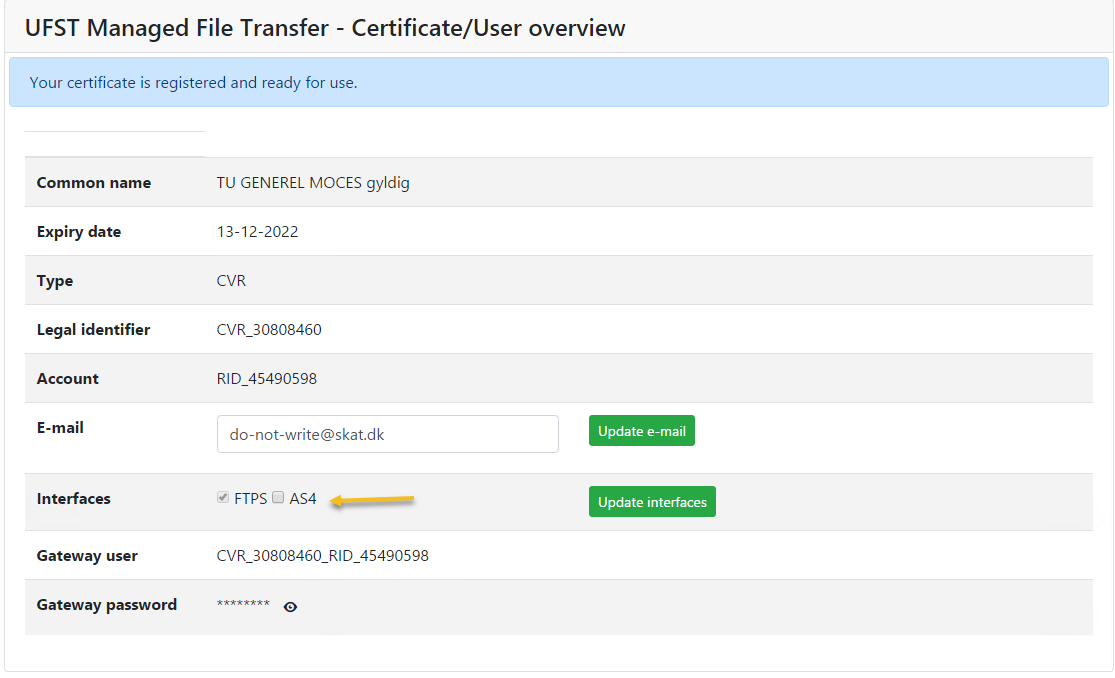
In this example the Selfservice portal has no previous knowledge of this certificate and would reject any logon attempt. This means that you must choose to “Register Certificate”



The registration process will be initiated and should be completed within a few minutes. Use the “Refresh” to verify when the registration has been completed.



The certificate is now registered, and you see both your Username and assigned password, which you should record for setup of your AS4 session.



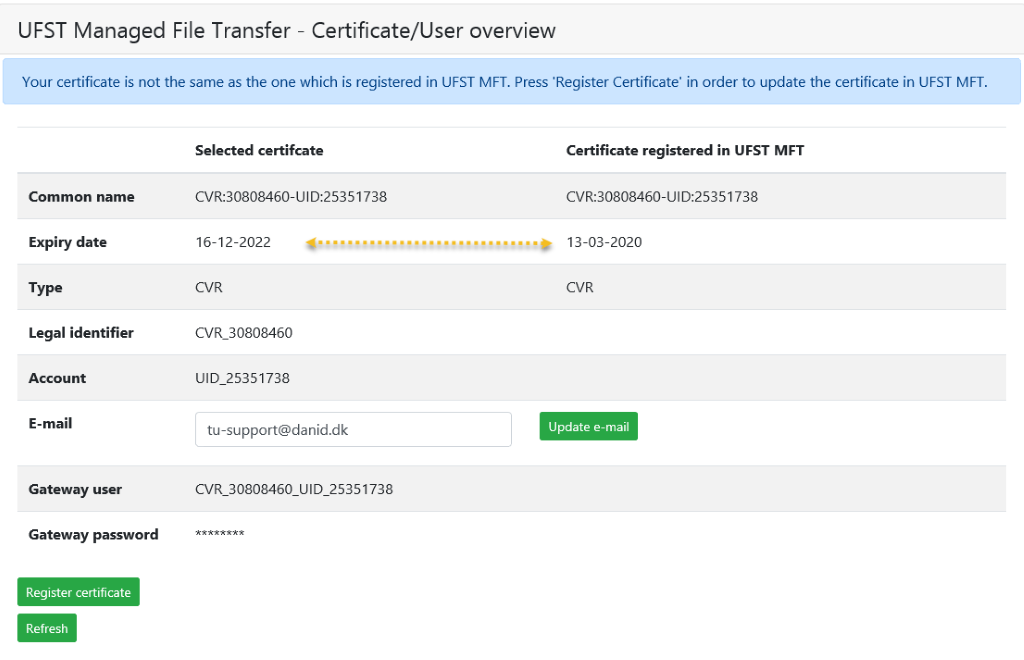
Note: Your email address is extracted from the certificate (if present). Please make sure you have a valid and relevant email address for your certificate as this could be used to contact you later.

Note: It is important that you checkmark the AS4 part in the interface section and click Update interfaces.

Finish by selecting “Log out”.  The FTPS Gateway login will be established within 15 minutes from your pre-registration, and you are then ready to upload to the services you have access to (verified with your DCS roles for certificate).

## Certificate Renewal

Whenever you renew a certificate (keeping the same UID/RID) you can use the Certificate Portal to update the certificate in FTPS Gateway. This is done by login on using your new certificate and the password you assigned during your first logon.



Use “Register Certificate” to update the certificate in FTPS Gateway.

Note: The procedure is the same as with a new certificate, except the assigned FTPS Gateway password will NOT change.

## Certificates

This section details how to acquire and setup the required certificates to authenticate a connection to DMS.

### User Certificate

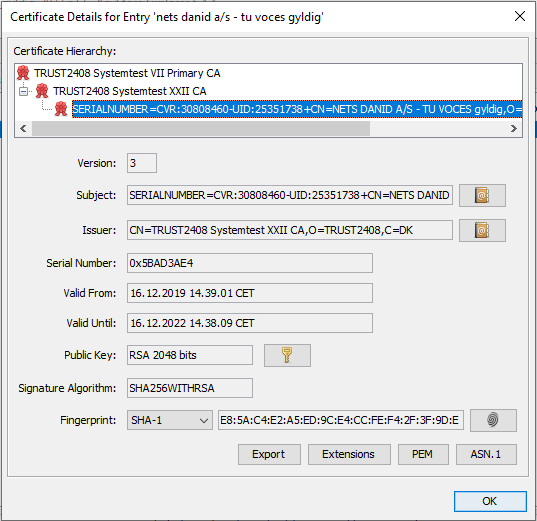
The user certificate used should be an already obtained VOCES certificate, if your company does not have a VOCES certificate, get in touch and we will help you with obtaining one.

### Correct conversion from .p12 to .jks certificate

VOCES certificates are usually distributed in .p12 file formats, and may need to be converted to .jks with the following command on Linux or Windows systems, with Java 8 or 11:

keytool -importkeystore -srckeystore '[VOCESCERTIFIKAT]' -srcstoretype pkcs12 -destkeystore privatekeys.jks -deststoretype JKS

Below is a screenshot of how the certificate should approximately look like in Keystore.



Note down and send CVR and UID to the team who provided you with this documentation.

# P-Modes

This section explains the P modes. P-Modes are used by some clients – eg. The holodeck and phase4 clients. The P-modes defines many of the parameters listed from the previous sections, in a single and movable file. The DMS solution only supports Push messages, and therefore this section only lists the P-mode allowing a push. The P-mode file, follows [this XSD](https://github.com/holodeck-b2b/Holodeck-B2B/blob/master/modules/holodeckb2b-core/src/main/resources/xsd/pmode.xsd). Additional details on the purpose, and usage on p-modes can be found here:

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4+-+1.14#eDeliveryAS4-1.14-P-ModeParameters>

A summation is provided below:

* An **Initiator** which contains details and identification of the initiating party for the transaction (in this case, the user contacting DMS)
* A **Responder** which contains details about the SecurityConfiguration of the recipient of the transaction.
* A **Leg** which contains information about the requested Protocol and Delivery as well as being the only major point of diversion between a Push and a Pull mode request.
* A push mode request contains a UserMessageFlow containing the required Action, and Service to be performed on the accompanying message.
* A pull mode request contains a PullRequestFlow.

## Declaration Push P-mode

On the next page you will find an example of the push p mode, with some values anonymized.  <PMode xmlns=**"http://holodeck-b2b.org/schemas/2014/10/pmode"**

    xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"** xsi:schemaLocation=**"http://holodeck-b2b.org/schemas/2014/10/pmode ../../repository/xsd/pmode.xsd"**>

<id include=**"false"**>**push-to-as4dms**</id>

<mep>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay**</mep>

<mepBinding>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push**</mepBinding>

<Initiator>

<PartyId type=**"string"**>**{Username}\_AS4**</PartyId>

<Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator**</Role>

<SecurityConfiguration>

<!-- Ensure that the authentication info is included in the sent messages -->

<UsernameToken target=**"ebms"**>

<username>**{Username}**</username>

<password type=**"text"**>**{UserPassword}**</password>

<includeNonce>**false**</includeNonce>

<includeCreated>**false**</includeCreated>

</UsernameToken>

<Signing>

<KeystoreAlias password=**"password"**>**{SigningPassword}**</KeystoreAlias>

<KeyReferenceMethod>**IssuerSerial**</KeyReferenceMethod>

</Signing>

</SecurityConfiguration>

</Initiator>

<Responder>

<PartyId type=**"string"**>**SKAT-MFT-AS4**</PartyId>

<Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder**</Role>

<SecurityConfiguration>

<!-- Ensure that the authentication info is included in the sent messages -->

<Encryption>

<KeystoreAlias>**secureftpgatewaytest.skat.dk**</KeystoreAlias>

</Encryption>

</SecurityConfiguration>

</Responder>

<Leg>

<Protocol>

<Address><**!—see endpoint section--><**Address>

</Protocol>

<DefaultDelivery>

<DeliveryMethod>**org.holodeckb2b.backend.file.NotifyAndDeliverOperation**</DeliveryMethod>

<Parameter>

<name>**format**</name>

<value>**ebms**</value>

</Parameter>

<Parameter>

<name>**deliveryDirectory**</name>

<value>**data/msg\_in**</value>

</Parameter>

</DefaultDelivery>

<UserMessageFlow>

<BusinessInfo>

<Action>**Declaration.Submit**</Action>

<Service>

<name>**DMS.Import**</name>

<type>**string**</type>

</Service>

</BusinessInfo>

<ErrorHandling>

<NotifyErrorToBusinessApplication>**false**</NotifyErrorToBusinessApplication>

</ErrorHandling>

</UserMessageFlow>

</Leg>

</PMode>

## Notification Push P-mode

And here is the notification push pmode

<PMode xmlns=**"http://holodeck-b2b.org/schemas/2014/10/pmode"**

xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"** xsi:schemaLocation=**"http://holodeck-b2b.org/schemas/2014/10/pmode ../../repository/xsd/pmode.xsd"**>

<id include=**"false"**>**push-noti-to-as4dms**</id>

<mep>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay**</mep>

<mepBinding>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push**</mepBinding>

<Initiator>

<PartyId type=**"string"**>**[Username]\_AS4**</PartyId>

<Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator**</Role>

<SecurityConfiguration>

<!-- Ensure that the authentication info is included in the sent messages -->

<UsernameToken target=**"ebms"**>

<username>**[Username]**</username>

<password type=**"text"**>**[UserPassword]**</password>

<includeNonce>**false**</includeNonce>

<includeCreated>**false**</includeCreated>

</UsernameToken>

<Signing>

<KeystoreAlias password=**"password"**>**dms-test-key**</KeystoreAlias>

<KeyReferenceMethod>**IssuerSerial**</KeyReferenceMethod>

</Signing>

</SecurityConfiguration>

</Initiator>

<Responder>

<PartyId type=**"string"**>**SKAT-MFT-AS4**</PartyId>

<Role>**http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder**</Role>

</Responder>

<Leg label=**"REQUEST"**>

<Protocol>

<Address>**http://localhost:8384/exchange/[Username]**</Address>

</Protocol>

<DefaultDelivery>

<DeliveryMethod>**org.holodeckb2b.backend.file.NotifyAndDeliverOperation**</DeliveryMethod>

<Parameter>

<name>**format**</name>

<value>**ebms**</value>

</Parameter>

<Parameter>

<name>**deliveryDirectory**</name>

<value>**data/msg\_in**</value>

</Parameter>

</DefaultDelivery>

<UserMessageFlow>

<BusinessInfo>

<Action>**Notification**</Action>

<Service>

<name>**DMS.Import**</name>

<type>**string**</type>

</Service>

</BusinessInfo>

<ErrorHandling>

<NotifyErrorToBusinessApplication>**true**</NotifyErrorToBusinessApplication>

</ErrorHandling>

</UserMessageFlow>

</Leg>

</PMode>

# XML Examples

*This section contains a few examples of properly formatted XML messages, with their AS4 headers, sent to DMS, with the replies received included. Fully signed message, with user-name and password:*



*DoubleClick to open the file*

## Synchronous answers example

This section contains two sets of examples of synchronous answers from Axway.

The structure of the ‘mi’ messages are mostly the same while the ‘pi’ messages contain a StatusResponse which carries a code and a message about the reception of the message sent in.

### Unapproved message

This set of answers are for an unapproved message, in this case the ‘pi’ answer contains information about the semantic mistakes that Axway detected in the message it is synchronously responding to in the format of all detected xml schema validation errors.



### Approved message

This set of answers is for an approved message, this ‘pi’ answer only contains a simple code (OK) which means that the message is approved and is currently being handled by the system. For further information a notification request should be sent to Axway which will then respond with processing notifications from the requested service for the accepted message.



# Possible Errors

This section contains the most common errors that have been reported by partners or observed internally when setting up a connection to DMS, with the goal of streamlining the setup as much as possible.

## DMS Fails to Authenticate User

Here is a possible list of reasons this could have happened:

* P mode misconfiguration:
* If the XML document is misconfigured DMS might refuse to authenticate the user, please consult the [XSD](https://github.com/holodeck-b2b/Holodeck-B2B/blob/master/modules/holodeckb2b-core/src/main/resources/xsd/pmode.xsd):
* Failure:        EBMS:0004 - Other - Unable to identify Party specified by From PartyId element(s).
* Initiator Party ID is wrong
* Failure:        EBMS:0004 - Other - Error in getting password for user [USERNAME]. User, password or policy is not valid or has expired or has been disabled.
* Username is wrong
* Password is wrong
* Massage failed to send, no Pmode found for message
* Signing Keystore Password is wrong
* Signing KeyReference Method is wrong.
* Keystore Alias is wrong.
* Failure:        EBMS:0004 - Other - Unable to identify Party specified by To PartyId element(s).
* Responder Party ID is wrong
* Message doesn't show up in DMS
* Protocol URL is wrong
* Failure:        "[ROUTING ID]" is an unknown routing id.
* Routing ID is wrong, right now the standard routing ID is: "DMS.Import.Declaration.Submit"
* Service is “DMS.Import”
* Action is “Declaration.Submit"

## Certificate Errors

These errors were obtained while trying to get a certificate from the [test certificate portal](https://secureftpgatewaytest.skat.dk/):

* “The supplied certificate is not a valid certificate” when accessing certificate portal
* The certificate is wrong, make sure you are using the correct VOCES certificate.
* “Login failed - verify that your password is correct”
* Certificate password for VOCES certificate is wrong, make sure you have the correct certificate password.

#Decorative

|  |  |  |  |
| --- | --- | --- | --- |
|  | www.ufst.dk  ISBN |  | Udviklings- og Forenklings- styrelsen er en del af  Skatteforvaltningen |