ICT&L REVENUE

Revenue
Cáin agus Custaim na hÉireann
Irish Tax and Customs

PAYE Modernisation

SOAP Connectivity Handshake Guide

Contents

Version	**1.0**
Version Date	**16/11/2018**

Version History			
Version	**Change Date**	**Section**	**Change Description**
V 1.0	30/10/2018	All	Document published.
	16/11/2018	4.2.2	Date timeout changed from 60 seconds to 90 minutes

Audience

This document is for any software provider who has chosen to build or update their products to allow for PAYE Modernisation.

Document context

This document provides a technical overview of how to integrate with Revenue's SOAP web services including how to sign requests validly. This document is designed to be read in conjunction with the Revenue Commissioners PAYE Modernisation documentation suite including the relevant technical documents.

Document References	
Reference	**Document Link**
1. Documents Homepage	Documents Homepage

1. Introduction

This document details the XML PAYE Modernisation web services specification for the following web services:

- SOAP Connectivity Handshake

The Documents Homepage specified in Document References is the home to all technical documentation, specification, and examples for the above web services which has been made available to enable payroll software developers to update their software packages to be compatible with PAYE reporting obligations from January 2019. Path locations specified in this document are relative to this Homepage.

This document assumes familiarity with the XML web services above. A full description of each of these can be found in the Web Service Overview Document under 'PAYE Web Service Examples' on the Documents Homepage. The Web Service Description Language (WSDL) files for the above web services can be found under 'PAYE Web Service Specifications'.

WSDL is a W3C standard for describing web services. Further details of each web service can be found in the supporting WSDL file. Each file references the necessary schemas and indicates the URL where the services may be accessed. The URL for each web service will use the HTTPS protocol to ensure the privacy of all communication between ROS and the web service client.

2. Calling the Services

The web services for the PAYE Modernisation messages are described through WSDL files and the schema for each message.

Further details of the web services can be found in the published PAYE Modernisation WSDL files. The WSDL, SOAP and WS Security version specifications we are following include:

- WSDL 1.2: WSDL 1.2
- SOAP 1.2: SOAP 1.2
- WS Security 1.1.1: WS Security 1.1.1

The SOAP Header must include a Content-Type of application/soap+xml.

2.1. Namespaces, Schemas and Locations

The PAYE Modernisation web services use namespaces which are detailed below.

2.1.1 Connectivity Handshake

The namespace defining all elements related to this web service is outlined in the following table:

Description	Namespace	Relative Location
Connectivity Handshake Web Service	N/A	TBC
Connectivity Handshake Request & Response Schema	http://www.ros.ie/schemas /paye/handshake/	TBC

2.2. Digital Signatures

The PAYE Modernisation web services will require a digital signature. This will be the digital signature of the declarant.

3. Interpreting the Responses

Each web service will return a response message to the client as outlined below.

3.1. Validation Errors

3.1.1. SOAP Faults

When a request is made to a PAYE Modernisation web service three checks are carried out before any processing can occur. These include:

1. Authentication

2. Authorisation

3. Schema validation

Step 1 of the validation process tries to verify the authenticity of the message by checking it has been signed with a valid digital signature. Step 2 focuses on authorising the credentials and finally, step 3 checks the message is validated against the schema.

Please note that Step 2 will only be performed if an Employer's PAYE tax registration number is provided as part of the request or if an Agent TAIN and Employer's PAYE tax registration number is provided as part of the request.

If there are any errors encountered carrying out the above processes then a SOAP fault with a HTTP status code of 500 - adhering to the Error Structure Guide - will be returned to the client. The fault string will provide more information on the details of the problem. Where a SOAP fault is returned to the client, no processing will occur for that message.

3.2. Successful Response

The "Connectivity" web service will return a ConnectionStatus of "SUCCESS" if called with a validly signed request and neither of the two optional parameters (EmployerRegistrationNumber, AgentTain) have been included in the request.

If the request includes an Employer's PAYE tax registration number (EmployerRegistrationNumber), the service will return a ConnectionStatus of "SUCCESS" if the request is validly signed and the provided tax registration number is the owner of the certificate that signed the request.

If the request includes both an Employer's PAYE tax registration number (EmployerRegistrationNumber) and Agent TAIN (AgentTain), the service will return a ConnectionStatus of "SUCCESS" if the request is validly signed, the provided Agent TAIN is the owner of the certificate and the provided tax registration number is linked to the Agent TAIN.

Please note that if an Agent TAIN is provided a linked Employer's PAYE tax registration number must also be provided.

A list of validation errors (if any) on the Connectivity Handshake Request is also included in the response. Please refer back to Section 3.1 for more information on Validation Errors.

4. Digital Signatures

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

The digital signature must be applied to the message in accordance with the WS-Security specification as specified in Section 4.1.

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

If a valid digital signature is not attached, a SOAP Fault will be returned. The fault string will provide more information on the details of the problem.

4.1. Namespaces

The valid approach for this is using Oasis standards:

- The WS-Security namespace should be: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
- The WSU namespace should be: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
- All Id references should now be of the form wsu: d e.g. <x:myElement wsu:Id="ID1"
- xmlns="…" xmlns:wsu="…"/>
- The XML Digital Signature namespace should be: http://www.w3.org/2000/09/xmldsig#

4.2. Security

The security header contains three elements:

1. The Binary Security Token
2. The Timestamp
3. The Signature.

4.2.1 Binary Security Token

The X509 certificate used to sign the message should be included in the message as a Base64 encoded BinarySecurityToken element (Envelope/Header/Security/BinarySecurityToken). 11 SOAP Web Service Integration Guide

The EncodingType attribute of the BinarySecurityToken should have a value of http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-soapmessage-security-1.0#Base64Binary.

The ValueType attribute should have a value of http://docs.oasis-open.org/wss/2004/01/oasis-200401- wss-x509-tokenprofile-1.0#X509v3.

4.2.2 Timestamp

The timestamp element (Envelope/Header/Security/Timestamp) must contain:

1. The Created date
2. The Expired date.

Both dates must conform to the following Oasis standard:

https://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-osSOAPMessageSecurity.htm#_Toc118717167 and the expired date must be after the created date.

For SOAP requests the maximum time between timestamp creation and expiration is 90 minutes. Please see sample Payroll Submission Valid Request Example.

4.2.3 Signature

This is the signature that is calculated using your ROS digital certificates private key. The signature contains three elements:

1. The SignedInfo element
2. The SignatureValue element
3. The KeyInfo element.

4.2.3.1 SignedInfo

The SignedInfo element contains a CanonicalizationMethod, SignatureMethod and two Reference elements. XML Canonicalization is used to format the XML before calculating the digest values. The SHA512withRSA algorithm is used for signing the message.

Canonicalization: The Canonicalization Algorithm should be XML-EXC-C14N (Exclusive Canonicalization) - Canonicalization Algorithm

Signature Algorithm: The Signature Algorithm should be SHA512withRSA - http://www.w3.org/2001/04/xmldsig-more#rsa-sha512

This type of message is known as a detached signature.

Oasis Standards References: There must be two Reference elements (Envelope/Header/Security/Signature/SignedInfo/Reference) in the SignedInfo element.

- One Reference element should correspond to the signed Body element within the message. The Reference corresponding to the Body should have an id attribute whose value is the same as the Id attribute of the SOAP element.
- The other Reference corresponds to the Timestamp. The Reference corresponding to the Timestamp should have an id attribute whose value is the same as the Id attribute of the SOAP element. Both References should have a single transform – Exclusive Canonicalization (see the URI above). The Digest Algorithm should be SHA512 - http://www.w3.org/2001/04/xmlenc#sha512.

4.2.3.2 SignatureValue

For the SignatureValue (Envelope/Header/Security/Signature/SignatureValue) extract the private key from the .p12 file. Please see Appendix A – Extracting from a .p12 File for instructions on how to do this.

4.2.3.3 KeyInfo

The KeyInfo contains a SecurityTokenReference element which contains a Reference corresponding to the BinarySecurityToken from Section 4.2.1 (Envelope/Header/Security/Signature/KeyInfo/SecurityTokenReference/Reference). The URI attribute of the Reference should reference the Id attribute of the BinarySecurityToken element. For example, if the Id attribute of the BinarySecurityToken is “X509Token”, the URI attribute of the Reference subelement should be “#X509Token”.

5. Example Messages

Message:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:han="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
   <soap:Header/>
   <soap:Body>
      <han:HandshakeRequest>
         <han:SoftwareUsed>
            <han:Name>SOAP</han:Name>
            <han:Version>1</han:Version>
         </han:SoftwareUsed>
      </han:HandshakeRequest>
   </soap:Body>
</soap:Envelope>
```

Expected Response:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns2:HandshakeResponse
xmlns:ns2="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
         <ns2:ConnectionStatus>SUCCESS</ns2:ConnectionStatus>
      </ns2:HandshakeResponse>
   </env:Body>
</env:Envelope>
```

Message:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:han="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
   <soap:Header/>
   <soap:Body>
      <han:HandshakeRequest>

<han:EmployerRegistrationNumber>1234567FA</han:EmployerRegistrationN
umber>
         <han:SoftwareUsed>
            <han:Name>SOAP</han:Name>
            <han:Version>1</han:Version>
         </han:SoftwareUsed>
```

```
      </han:HandshakeRequest>
   </soap:Body>
</soap:Envelope>
```

Expected Response:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns2:HandshakeResponse
xmlns:ns2="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
         <ns2:ConnectionStatus>SUCCESS</ns2:ConnectionStatus>
      </ns2:HandshakeResponse>
   </env:Body>
</env:Envelope>
```

Message:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:han="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
   <soap:Header/>
   <soap:Body>
      <han:HandshakeRequest>

<han:EmployerRegistrationNumber>1234567FA</han:EmployerRegistrationN
umber>
         <han:AgentTain>12345A</han:AgentTain>
         <han:SoftwareUsed>
            <han:Name>SOAP</han:Name>
            <han:Version>1</han:Version>
         </han:SoftwareUsed>
      </han:HandshakeRequest>
   </soap:Body>
</soap:Envelope>
```

Expected Response:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns2:HandshakeResponse
xmlns:ns2="http://www.ros.ie/schemas/paye-employers/v1/handshake/">
         <ns2:ConnectionStatus>SUCCESS</ns2:ConnectionStatus>
      </ns2:HandshakeResponse>
   </env:Body>
</env:Envelope>
```