IRmark generation – Step By Step

The IRmark process can be split into a few main steps. This document will attempt to give clear guidelines on these steps, but should be used in conjunction with the technical documents in this pack. This document will not cover the response schema and the legislative requirements around it.

Process 1 – Preparing the XML file

The IRmark is generated from the payload of the submission so this part of the XML must be extracted first. The payload is everything inside and **including** the <Body></Body> node. When you extract the body you must "inherit" any namespace declarations in the <GovTalkMessage> node and place them in the <Body> node. See Fig 1.

Fig 1.

<GovTalkMessage xmlns="http://www.govtalk.gov.uk/CM/envelope">

<Body xmlns="http://www.govtalk.gov.uk/CM/envelope">

N.B. This is just for the generation of the IRmark and the namespace should not be present in the <Body> on submission.

Finally, to prepare the XML you need to remove the IRmark node from the <Body>. However you choose to do this any data around the IRmark opening and closing tags e.g. white space, line-endings, tabs etc, must be preserved.

Process 2 – Canonicalisation

The extracted XML must then be canonicalised. The W3C spec for canonicalisation can be found here:

http://www.w3.org/TR/2001/REC-xml-c14n-20010315

It is not advisable to try and write your own component to achieve this. Most popular development environments will already have C14n components that will do this work for you. It is also possible to prepare your XML up front when it is generated so that it is essentially already canonicalised, however care should be taken when doing this to make sure you cater for character references and so forth. There is an example of canonical XML in the tech pack.

Process 3 – SHA-1 Hash

Once the XML is in canonical form a SHA-1 digest must be created from it. Again nearly every environment should have several components available that handle this. The important thing to remember is that your resulting SHA-1 hash must be in binary (160 bits long).

Process 4 – Base64/32 encoding

The SHA-1 hash can then be Base64 and Base32 encoded. As with the other two processes this should be done using already available components and the results should be printable/readable strings that can be inserted into the XML (Base64) and displayed to the user (Base32)

Things to Remember

The C14n specification does not require removal of white space from inside your <Body> node (apart from normalising line-endings to line feed characters). So the XML that you extract to generate the IRmark should be formatted in exactly the same way as the XML that is eventually submitted. We advise that your XML should have no extra spacing at all, it is not needed and will cut down on some of the problems you may find while implementing the IRmark. We also advise that you do not include comments in the body of your submission.

If you are using the sample Java code

When using the sample code in the tech pack, there are a couple of small changes that need to be made for it to work as it should.

1. Line 78:
   "<dsig:Transforms xmlns:dsig='http://www.w3.org/2000/09/xmldsig#' xmlns:gt='http://www.govtalk.gov.uk/CM/envelope' xmlns:ir='**http://www.govtalk.gov.uk/taxation/SA**'>\n

   The namespace highlighted above needs to be the same as the namespace in the <IRenvelope> node of your XML file.

2. Line 86:

   The algorithm should not be "WithComments". You just need to remove the "WithComments" part of this line.

If you are using the sample code, everything is handled for you, so you do not need to replicate any of the processes in this document.

If you have any suggestions for information that could go into this document to help with the understanding of the IRmark process please forward them to:

SDSTeam@ir.gsi.gov.uk