

IRmark Generation  
Step By Step Guide  
for Gateway Protocol based services

<b>1. Generating the IRmark</b> .....	<b>3</b>
1.1 Preparing the XML file .....	3
1.2 Canonicalisation .....	4
1.3 SHA-1 Hash .....	4
1.4 Base64/32 encoding .....	4
<b>2. General Advice</b> .....	<b>5</b>
2.1 White Space .....	5
2.2 Comments in the payload .....	5
2.3 If you are using the sample Java code .....	5

## 1. Generating the IRmark

If you are generating an IRmark for a SOAP based service please see the step by step guide for SOAP services.

The IRmark process can be split down into a few main steps. This document will attempt to give clear guidelines on these steps. This document will not cover the response schema and the legislative requirements around the IRmark.

### 1.1 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 and all** 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. The above is for illustrative purposes only, if you have other namespaces declared in the GovTalkMessage these will also need to be inherited.

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**.

## 1.2 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 are examples of canonical XML in the tech pack.

## 1.3 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 or in other words an array of 20 bytes)

## 1.4 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)

## 2. General Advice

### 2.1 White Space

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, as well as reducing file size and processing time.

### 2.2 Comments in the payload

**Do not** use comments in the body of your submission. It may be impossible to avoid the use of comments for Corporation Tax submissions that include iXBRL, in which case we advise that you Base64 encode the iXBRL before IR mark calculation and file transmission.

### 2.3 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/xmlldsig#'
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.

There is also an executable JAR in the techpack that will work for all Gateway Protocol based services, please see the readme included in the archive for instructions on how to use it.

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@hmrc.gsi.gov.uk](mailto:SDSTeam@hmrc.gsi.gov.uk)