Redirecting

```
Michael Edge wrote:

> It seems to me that the following code will not work in Cocoon 2.
> I believe it works in Cocoon 1, but for some reason the request
> and response objects available within XSP for Cocoon 2 are not
> HttpServletRequest and HttpServletResponse objects, but rather
> some cocoon equivalents, org.apache.cocoon.environment.Response
> and org.apache.cocoon.environment.Request. Unfortunately these
> new classes do not include the sendRedirect method. Any ideas
> how I would do this? I need to construct a URL in my code
> and redirect to that URL.

  <xsp:logic>
    String url = URLEncoder.encode(url + whatever + parameters);
    response.sendRedirect(url);
  </xsp:logic>
```

Reply:

Don't use XSPs for flow control, use the sitemap. This is a common error with new users, that forget that it's the sitemap that manages the relations between pages.

If it's in the pages, the "flow" logic is dispersed; in the Cocoon sitemap it's easier to see, manage and change.

Even more so, sending a HTTP redirect works only reliably before any content has been delivered to the client. Because of Cocoon's pipeline model this is almost impossible to do from a XSP.

Add an Action that does this and redirect. You can write XSPAction BTW.

Better still, use the action to return the url and then do a <Redirects> in the sitemap; it makes your action more reusable.

Another reply:

I wanted to redirect according to a URL provided in one of the XML documents being processed; something like this:

```
  <map:match pattern="my-resource">
    ...
    <map:transform type="special-transformer"/>
    <map:redirect-to uri="{request-attr:my-parameter}"/>
  </map:match>
```

Here, the "special-transformer" would be based on AbstractDOMTransformer and set request attributes when it saw various things in the document. Unfortunately, the redirect is "precalculated" and the desired value for "my-parameter" never gets used.

So what I did instead was to use some arguably nasty code in my "special-transformer" to set the redirect:

```
import org.apache.cocoon.environment.Response;
import org.apache.cocoon.environment.http.HttpResponse;

  // ...

  if (response instanceof HttpResponse) {
    ((HttpResponse) response).sendPermanentRedirect(target);
  }
```

This isn't nice, but there really doesn't seem to be any sane way in "vanilla" Cocoon where sitemap actions can be influenced by the content being processed. I don't want to use FlowScript, XSP or any of the peripherals.

Yet another reply:

The redirect attempted by the previous reply can be done using the cinclude transformer. In your pipeline, this looks like:

```
        <map:transform src="redirect-include.xsl">
          <map:parameter name="uri" value="{request-attr:my-parameter}"/>
        </map:transform>
        <map:transform type="cinclude"/>
```

The stylesheet 'redirect-include.xsl' is simply:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>

  <xsl:param name="uri"/>

  <xsl:template match="/">
    <ci:include src="{$uri}" xmlns:ci="http://apache.org/cocoon/include/1.0"/>
  </xsl:template>

</xsl:stylesheet>
```

After cinclude has done its work, the rest of the original pipeline is executed, so you will need at least a serializer after cinclude.