

Greenstone 3

Skins++ Technical Documentation

In the aim of simplifying the transformation XSLT files, we have added another layer of abstraction. In other words, we have added a library file containing the XSLT code for the most commonly used elements, and this code is made available in each transformation .xsl files with new elements called GS-Lib statements.

We have tried to use this technique to allow simplification of each transformation file so that they will become similar to HTML documents, thereby allowing more people to understand them.

Now each interface needs to use two new files inside the transform folder:

- **library.xsl** is the place where the user can set the doctype of the HTML pages produced. This is also the place where you can create personalised GS-Lib elements that could be used in any transformation page. The content of **util.xsl** is imported into this style sheet, this is required.
- **util.xsl** contains all the code provided by us to expand all GS-Lib statements, this is the code previously used in style.xsl.

Moreover all interfaces will now be pre-processed by:

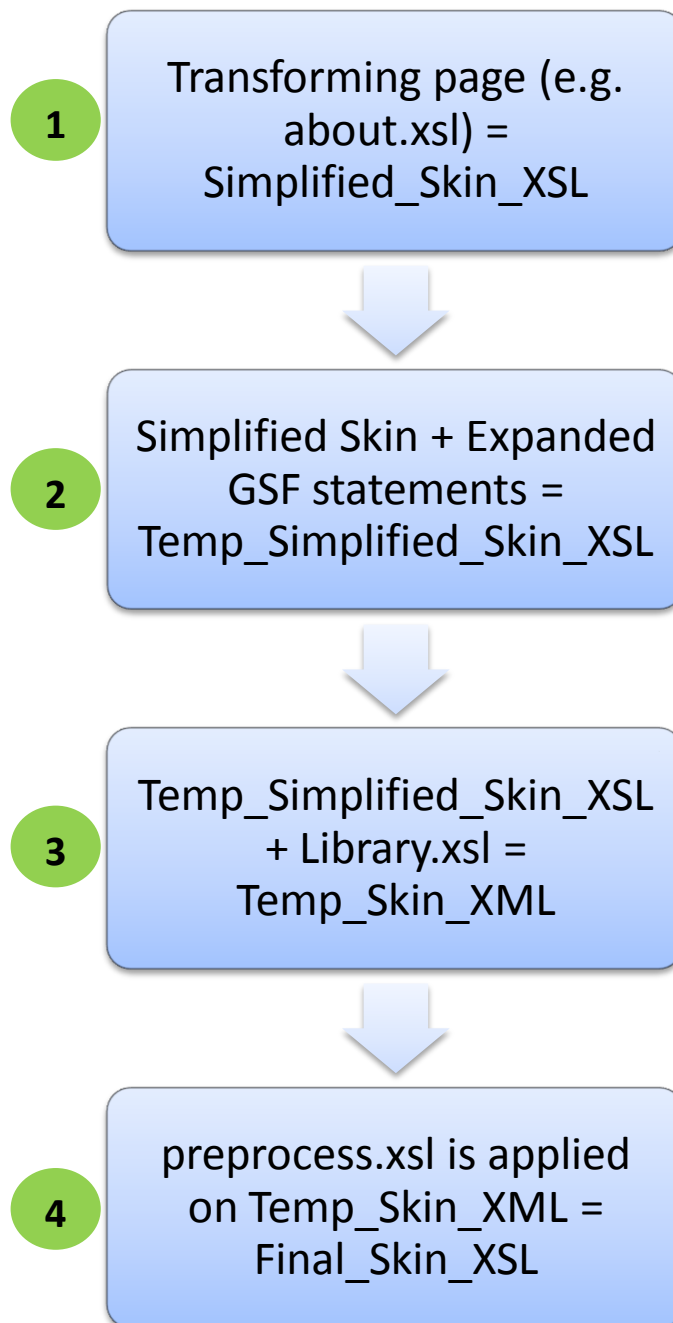
- **preprocess.xsl**, which will expand all GS-Lib statements from a simplified transformation page using the code coming from library.xsl. (In fact, the GS-Lib statement will be replaced by an “apply-template” of the same name located inside library.xsl or util.xsl.)

We can find preprocess.xsl in greenstone3/web/ui/xslt/preProcess.xsl

Any new GS-Lib statement created in util.xsl or library.xsl will automatically be available in any transformation page without having to restart anything: just reload the web page and it should work.

The transformation page containing the GS-Lib statements is pre-processed before being used to transform the XML data coming from the Message Router to a final HTML page. Everything happens in memory just before each web page is generated. To see the code, have a look at the file `greenstone3/src/java/org/greenstone/gsd3/core/TransformingReceptionist.java`.

The simple diagram below outlines the process:



- **Step 1** is really the content of a transformation file such as home.xsl or about.xsl
- **Step 2** can be seen during post-processing using the (experimental) debug statement `&o=skindoc` at the end of an URL.
- **Step 3** is an XML document with a root element called **skinAndLibraryXsl** and only two children (**skinXsl** and **libraryXsl**). We can see this temporary step using the (experimental) debug statement `&o=skinandlib` at the end of an URL.
- **Step 4** is the final XSL document that will be applied to the XML data. Type `&o=skinandlibdoc` to display it in the browser. Don't forget that `&o=xm1` will display the information coming from the Message Router.