Hello from the Sun `xmlroff` XSL Formatter

Russian	Здравствуйте!	Hebrew	שלום	Greek	Γειά σας
S. Chinese	你好	Japanese	こんにちは	Korean	안녕하세요, 안녕하십니까

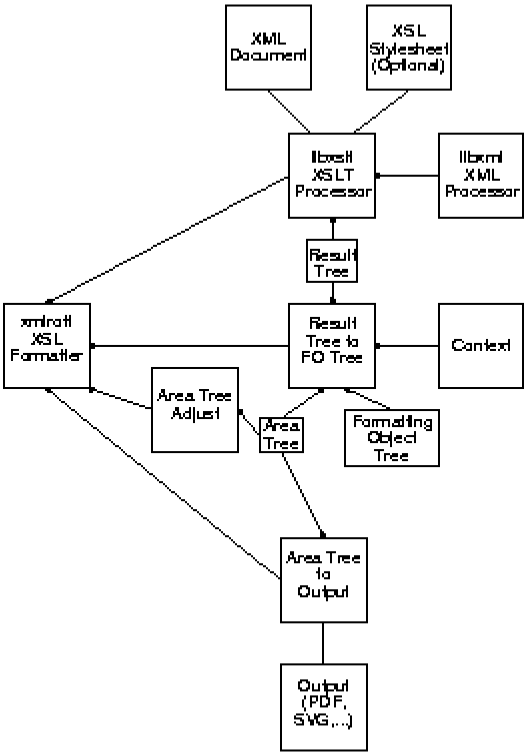
The Sun `xmlroff` XSL formatter is written in C, and uses it uses libxml2 and libxslt plus the GLib, GObject and Pango libraries that underlie GTK+ and GNOME (although it does not require either GTK+ or GNOME). GLib is a general-purpose utility library, GObject is a flexible extensible object-oriented framework for C, and Pango is a framework for the layout and rendering of internationalized text. This combination made it easier to develop the formatter, makes it easier for current GTK+ and GNOME developers to also work on the formatter, and allows the formatter to use the internationalization support of Pango

`xmlroff` currently produces PDF output using the PDFlib library. Other output formats can be added.

`xmlroff` is a command line program, but the bulk of the XSL formatting is implemented as a `libfo` library that can be linked with any program that requires XSL formatting capability.

Context Diagram

The context diagram is in the form used in *Software Requirements & Specifications* by Michael Jackson.



Source XML to Result Tree. This stage uses libxml2 and libxslt to transform the source XML and, optionally, XSL stylesheet into an in-memory representation of the formatting objects and their properties that are used to direct the formatting.

Result Tree to Formatting Object and Area Trees. This stage transforms the result tree into a tree of real programmatic objects with properties that are expressed as numeric, boolean, color, or other datatypes (instead of just text) and creates a tree of objects representing the formatted document layed out onto pages. `xmlroff` builds the formatting object tree and the area tree in parallel so expressions containing percentages are resolved when a formatting object is added to the formatting object tree.

Area Tree Adjustment. This stage works on the area tree as a whole to optimize the arrangement of the areas.

Creating each page in isolation produces a workable result, but there can be dependencies between pages; for example, page number citations to other pages. In addition, producing quality pages (i.e., pages that look good) means, for example:

- Balancing the amount of text on facing pages so the content of both pages extend the same distance down the page
- Aligning the lines on facing pages and on back-to-back pages
- Not splitting a block of text such that only one line appears before a page break or only one line appears after a break
- Not ending a page on a hyphen

Area Tree to Output. This stage writes out the area tree in a format that can be used by other programs or sent to a printer. The initial output format is PDF.