

# ErlGuten

*Using Erlang in typography applications*

*ErlGuten.* is a system for high quality typesetting, ErlGuten is free software. ErlGuten aims to produce typographic quality PDF directly from XML or from a program.

The aim of ErlGuten is to produce high quality PDF from a layout language or from a program. The ErlGuten distribution includes a programming API, so that Erlang programs can produce PDF - and a typesetting system for typesetting documents written in XML.

The document that you are reading was authored in XML and typeset by ErlGuten. The name ErlGuten is chosen because the program is written in Erlang - the *Guten* part is a reference to *Gutenberg* the father of printing.

ErlGuten is a system for high quality typesetting, so we take a great deal of care when formatting text, a large number of optimizations are performed which improve the quality of the printed text. Many of these optimizations are usually only found in expensive professional type-setting programs. We believe that WYSIWYG programs have destroyed the fine art of typesetting - ErlGuten is a modest attempt to improve the situation.

We have chosen XML as the input language for its wide appeal. XML provides only a thin abstraction layer over the typesetting system - so the adventurous can use the programming interface to ErlGuten - to directly produce typographic quality PDF in real-time. We expect this facility to be useful for the dynamic generation of documents from web-servers.

In ErlGuten we take the view that the highest level of abstraction is the *layout* of a document - we are very concerned that the user can specify the *exact* position of text on the printed page. At the next level of abstraction we are concerned with the typefaces that are used to format different regions of the document.

ErlGuten is designed for the production of large and complex documents with complex layout requirements, like *newspapers* or *books*. In ErlGuten layout, content, and document management are considered separate issues. Layout is *template based* - Content is assumed to be stored as a large number of documents in a file system or data base, document management is considered as a mapping operation which takes documents in the content data base and maps them onto templates to produce high quality output.

This is normal text, set 30 picas wide in 12/14 Times Roman. Many different typefaces can be used within a paragraph. *Emphasized text* is set in Times-Italic. Hyphenation uses the TeX hyphenation algorithm. Any of the 35 built-in PDF typefaces with the same point size can be mixed with a paragraph. The term `{person, "Joe"}` is an Erlang term which has been typeset in 12 point *courier*. The paragraph justification algorithm does proper *kerning* so, for example, the word AWAY is correctly kerned! - line breaks within a paragraph are selected by a sophisticated global optimization technique.