NGerlguten -- Next Generation Erlguten

The erlguten process can be called to process XML and galley files into PDF output. This output can either be read from a file and written to an output file or processed as a input XML string with a PDF string returned.

Erlguten.erl is called the following ways.

`erlguten:format(Pid, File)`	returns the results to an output file
`erlguten:format_string(Pid, String, Root)`	returns the results as a PDF binary (ready to write out).
`Erlguten:dispatch(Pid, File, To)`	Puts the PDF results in a file and sends **ok** to the To process.
`Erlguten:dispatch_string(Pid, String, Root, To)`	Delivers the PDF binary to the To process.

Pid is the process ID for the instance of erlguten.

File is a filename with an xml extension. It is located in the same folder as the galley files (.gal extension). The File is read, parsed and processed according to the galley files into an output file in the same folder with the same name as the XML input with a .pdf output file extension. For example, “C:/doco/doc1.xml” creates “C:/doco/doc1.pdf”.

String is an Erlang list holding the text that would be in a file to process in the erlguten:format call.

Root is the name of the folder holding the galley files when you are processing a String instead of a file. If the galley file is “/usr/local/gal/galley001.gal”, the Root value is “/usr/local/gal”.

To is the PID of a process to receive the results of processing a PDF. For dispatch the results on success is the atom ok. For dispatch_string, the binary Pdf results is sent to the process.