erl2latex PDF Build Script

Ulf Wiger

March 16, 2009

Copyright (c) 2008 Ulf Wiger, John Hughes[1]

```erlang
#!/usr/bin/env escript
%%! -smp enable


-include_lib("kernel/include/file.hrl").

main(Args) ->
    try code:add_path(path()),
        chk_args(Args) of
        {F, Opts} ->
            case erl2pdf:file(F, Opts) of
                ok ->
                    return_ok();
                _Error ->
                    rpt("Conversion to PDF failed.~n", []),
                    return_error()
            end;
        OtherArgs ->
            rpt("Bad args (~p).~n"
                "Usage: make_pdf Src Target~n", [OtherArgs]),
            return_error()
    catch
        error:E ->
            rpt("Exception (~p)~n", [E]),
            return_error()
    end.

 chk_args(Args) ->
    chk_args(Args, {undefined,[]}).

 chk_args(["-o",Out|Args], {Src,Opts}) ->
    chk_args(Args, {Src,[{outdir,Out}|Opts]});
```

```
chk_args(["-keep_temp"|Args], {Src, Opts}) ->
    chk_args(Args, {Src, [keep_temp|Opts]});
chk_args([Src|Args], {_,Opts}) ->
    chk_args(Args, {Src,Opts});
chk_args([], {Src,Opts}) ->
    {Src,Opts}.

rpt(Msg,Args) ->
    io:fwrite("make_pdf: " ++ Msg, Args).

return_error() ->
    init:stop(-1).

return_ok() ->
    init:stop(0).
```

This fairly elaborate function tries to find the ebin/ directory where erl2latex.beam resides. It is designed to work even if the make_pdf script is started from a symlink to the original.

```
path() ->
    Script = escript:script_name(),
    Src = src_name(Script),
    Abs = filename:absname(Src),
    filename:join(dir(dir(Abs)), "ebin").

src_name(F) ->
    case file:read_link_info(F) of
        {ok, #file_info{type = symlink}} ->
            {ok, Src} = file:read_link(F),
            src_name(Src);
        {ok, #file_info{type = regular}} ->
            F
    end.

dir(F) ->
    filename:dirname(F).
```