```
   Help
#include  "hes1d_std.h"
#include  "math/equity_pricer/levy_diffusion.h"
#include  "math/equity_pricer/carr.h"
#include  "std/std.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion <
     (2010+2) //The "#else" part of the code will be freely av
    ailable after the (year of creation of this file + 2)
static int CHK_OPT(CF_CarrHeston)(void *Opt, void *Mod)
{
  return NONACTIVE;
}
int CALC(CF_CarrHeston)(void*Opt,void *Mod,PricingMethod *
    Met)
{
return AVAILABLE_IN_FULL_PREMIA;
}
#else

int CALC(CF_CarrHeston)(void *Opt, void *Mod, Pricing
    Method *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;
  NumFunc_1 *p;
  int option_type;
  int std=1;
  if(ptMod->Sigma.Val.V_PDOUBLE==0.0)
    {
      Fprintf(TOSCREEN,"BLACK-SHOLES MODEL{n{n{n");
      return WRONG;
    }
  else
    {
      double drift;
      Option_Eqd *op;
      Heston_diffusion *Process= Heston_diffusion_create(pt
    Mod->LongRunVariance.Val.V_PDOUBLE,
                                                         pt
    Mod->MeanReversion.hal.V_PDOUBLE,
```

```
                                                                pt
    Mod->Rho.Val.V_PDOUBLE,
                                                                pt
    Mod->Sigma.Val.V_PDOUBLE,
                                                                sq
    rt(ptMod->Sigma0.Val.V_PDOUBLE),
                                                                &
    drift);
      Levy_diffusion * Levy =Levy_diffusion_create(Process,
    &Heston_diffusion_characteristic_exponent,&Heston_diffusio
    n_ln_characteristic_function);
      p=ptOpt->PayOff.Val.V_NUMFUNC_1;
      if ((p->Compute)==&Call)
        option_type=1;
      else
        if((p->Compute)==&Put)
          option_type=2;
        else
          option_type=3;

      op=option_eqd_create(ptOpt->EuOrAm.Val.V_BOOL,option_
    type,std,ptMod->S0.Val.V_PDOUBLE,p->Par[0].Val.V_DOUBLE,pt
    Opt->Maturity.Val.V_DATE-ptMod->T.Val.V_DATE,0,0);
      option_eqd_set_rate(op,log(1.+ptMod->R.Val.V_DOUBLE/1
    00.),log(1.+ptMod->Divid.Val.V_DOUBLE/100.));


      CarrMethod_Vanilla_option_LD(op,0.1,Levy);
      (Met->Res[0].Val.V_DOUBLE)=op->price;
      (Met->Res[1].Val.V_DOUBLE)=op->delta;
      free(op);
      free(Levy);
      free(Process);
      return OK;
    }
}

static int CHK_OPT(CF_CarrHeston)(void *Opt, void *Mod)
{
  if ((strcmp( ((Option*)Opt)->Name,"CallEuro")==0)||(strc
    mp( ((Option*)Opt)->Name,"PutEuro")==0))
```

```
    return OK;

  return  WRONG;
}
#endif //PremiaCurrentVersion

static int MET(Init)(PricingMethod *Met,Option *Opt)
{
  if ( Met->init == 0)
    {
      Met->init=1;
    }

  return OK;
}

PricingMethod MET(CF_CarrHeston)=
{
  "CF_Carr_Heston",
  {{" ",PREMIA_NULLTYPE,{0},FORBID}},
  CALC(CF_CarrHeston),
  {{"Price",DOUBLE,{100},FORBID},
   {"Delta",DOUBLE,{100},FORBID} ,
   {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CHK_OPT(CF_CarrHeston),
  CHK_ok,
  MET(Init)
};
```

References