```
   Help
#include  "doublehes1d_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_CarrDoubleHeston)(void *Opt, void *
    Mod)
{
  return NONACTIVE;
}
int CALC(CF_CarrDoubleHeston)(void*Opt,void *Mod,Pricing
    Method *Met)
{
return AVAILABLE_IN_FULL_PREMIA;
}
#else

int CarrDoubleHeston(double S,NumFunc_1  *p, double T,
    double r, double divid, double z1,double z2,double z3,double k,
    double c,double sigma1,double sigma2,double rho1,double rho2,
    double rho3,double *ptprice,double *ptdelta)
{
  int flag_call;
  double K,prix=0.,delta=0.;

  K=p->Par[0].Val.V_PDOUBLE;

  if ((p->Compute)==&Call)
    flag_call=1;
  else
    flag_call=0;;

  /* Price*/
  *ptprice=prix+K;

  /* Delta */
  *ptdelta=delta+flag_call;

  return OK;
}
```

```
int CALC(CF_CarrDoubleHeston)(void *Opt, void *Mod, Pricing
    Method *Met)
{
  TYPEOPT* ptOpt=(TYPEOPT*)Opt;
  TYPEMOD* ptMod=(TYPEMOD*)Mod;
  double r,divid;

  if(ptMod->Sigma.Val.V_PDOUBLE==0.0)
    {
      Fprintf(TOSCREEN,"BLACK-SHOLES MODEL{n{n{n");
      return WRONG;
    }
  else
    {
      r=log(1.+ptMod->R.Val.V_DOUBLE/100.);
      divid=log(1.+ptMod->Divid.Val.V_DOUBLE/100.);

      return CarrDoubleHeston(ptMod->S0.Val.V_PDOUBLE,
        ptOpt->PayOff.Val.V_NUMFUNC_1,
        ptOpt->Maturity.Val.V_DATE-ptMod->T.Val.V_DATE,
        r,
                              divid, ptMod->Sigma0.Val.V_
    PDOUBLE,
                              ptMod->Sigma0V.Val.V_PDOUBLE,
                               ptMod->LongRunVarianceV.Val.
    V_PDOUBLE
                              ,ptMod->MeanReversion.hal.V_
    PDOUBLE
                              ,ptMod->MeanReversionV.Val.V_
    PDOUBLE,
                              ptMod->Sigma.Val.V_PDOUBLE,
                                ptMod->SigmaV.Val.V_PDOUB
    LE,
                              ptMod->Rho.Val.V_DOUBLE,
                              ptMod->RhoSV2.Val.V_DOUBLE,
                               ptMod->RhoVV.Val.V_DOUBLE,
        &(Met->Res[0].Val.V_DOUBLE),
        &(Met->Res[1].Val.V_DOUBLE)
        );
    }
}


static int CHK_OPT(CF_CarrDoubleHeston)(void *Opt, void *
    Mod)
{
  return NONACTIVE;
  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_CarrDoubleHeston)=
{
  "CF_Carr_DoubleHeston",
  {{" ",PREMIA_NULLTYPE,{0},FORBID}},
  CALC(CF_CarrDoubleHeston),
  {{"Price",DOUBLE,{100},FORBID},
   {"Delta",DOUBLE,{100},FORBID} ,
   {" ",PREMIA_NULLTYPE,{0},FORBID}},
  CHK_OPT(CF_CarrDoubleHeston),
  CHK_ok,
  MET(Init)
};
```

References