Help

```
#include "hullwhite1d_stdi.h"

//The "#else" part of the code will be freely available after the (year of creat
#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2007+2)
static int CHK_OPT(CF_ZCBondHW1D)(void *Opt, void *Mod)
{
  return NONACTIVE;
}
int CALC(CF_ZCBondHW1D)(void *Opt, void *Mod, PricingMethod *Met)
{
  return AVAILABLE_IN_FULL_PREMIA;
}
#else


void ZCPrice_CoefficientHW1D(ZCMarketData *ZCMarket, double a, double sigma, dou
{
  double f0_t, P0_t, P0_T, P0_t_plus, P0_t_minus;

  /*Computation pure discount bond*/
  P0_t = BondPrice(t, ZCMarket);
  P0_T = BondPrice(T, ZCMarket);

  /*Computation of Forward rate*/
  P0_t_plus = BondPrice(t + INC, ZCMarket);
  P0_t_minus = BondPrice(t, ZCMarket);
  f0_t = -(log(P0_t_plus) - log(P0_t_minus)) / (INC);

  /*A,B coefficient*/
  (*B_tT) = (1. / a) * (1. - exp(-a * (T - t)));

  (*A_tT) = (P0_T / P0_t) * exp((*B_tT) * f0_t - (sigma * sigma / (4.*a)) * (1.
}

// Price of a ZC using the three coefficient A(t,T), B(t,T) and C(tT). H&W is a
double ZCPrice_Using_CoefficientHW1D(double r_t, double A_tT, double B_tT)
{
  return A_tT * exp(-B_tT * r_t);
}
```

```
// Price at date t of a ZC maturing at T, knowing that r(t)=r_t and u(t)=u_t.
double cf_hw1d_zcb(ZCMarketData *ZCMarket, double a, double sigma, double t, dou
{
  double price;
  double A_tT, B_tT;
  A_tT = 0;
  B_tT = 0;

  ZCPrice_CoefficientHW1D(ZCMarket, a, sigma, t, T, &A_tT, &B_tT);
  price = ZCPrice_Using_CoefficientHW1D(r_t, A_tT, B_tT);

  return price;

}

static int cf_zcbond1d(int flat_flag, double r_t, char *curve, double a, double
{

  ZCMarketData ZCMarket;

  /* Flag to decide to read or not ZC bond datas in "initialyields.dat" */
  /* If P(0,T) not read then P(0,T)=exp(-r0*T) */
  if (flat_flag == 0)
    {
      ZCMarket.FlatOrMarket = 0;
      ZCMarket.Rate = r_t;
    }

  else
    {
      ZCMarket.FlatOrMarket = 1;
      ZCMarket.filename = curve;
      ReadMarketData(&ZCMarket);

      r_t = -log(BondPrice(INC, &ZCMarket)) / INC;

      if (T > GET(ZCMarket.tm, ZCMarket.Nvalue - 1))
        {
          printf("\ nError : time bigger than the last time value entered in ini
          exit(EXIT_FAILURE);
```

```
        }
    }

  //Price of an option on a ZC
  *price = cf_hw1d_zcb(&ZCMarket, a, sigma, 0, r_t, T);

  DeleteZCMarketData(&ZCMarket);

  return OK;
}


int CALC(CF_ZCBondHW1D)(void *Opt, void *Mod, PricingMethod *Met)
{
  TYPEOPT *ptOpt = (TYPEOPT *)Opt;
  TYPEMOD *ptMod = (TYPEMOD *)Mod;


  return cf_zcbond1d(ptMod->flat_flag.Val.V_INT,
                     MOD(GetYield)(ptMod),
                     MOD(GetCurve)(ptMod),
                     ptMod->a.Val.V_DOUBLE,
                     ptMod->Sigma.Val.V_PDOUBLE,
                     ptOpt->BMaturity.Val.V_DATE - ptMod->T.Val.V_DATE,
                     &(Met->Res[0].Val.V_DOUBLE));
}
static int CHK_OPT(CF_ZCBondHW1D)(void *Opt, void *Mod)
{
  return strcmp(((Option *)Opt)->Name, "ZeroCouponBond");
}
#endif //PremiaCurrentVersion



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

```
  return OK;
}

PricingMethod MET(CF_ZCBondHW1D) =
{
  "CF_HullWhite1d_ZCBond",
  {{" ", PREMIA_NULLTYPE, {0}, FORBID}},
  CALC(CF_ZCBondHW1D),
  {{"Price", DOUBLE, {100}, FORBID}/*,{"Delta",DOUBLE,{100},FORBID} */, {" ", PR
  CHK_OPT(CF_ZCBondHW1D),
  CHK_ok,
  MET(Init)
} ;
```