

[Help](#)

```

extern "C" {
#include "hes1d_vol.h"
#include "numfunc.h"

}

extern "C" {

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2008+2) //The "#els
    static int CHK_OPT(CF_HES_VARIANCESWAP)(void *Opt, void *Mod)
    {
        return NONACTIVE;
    }
    int CALC(CF_HES_VARIANCESWAP)(void *Opt, void *Mod, PricingMethod *Met)
    {
        return AVAILABLE_IN_FULL_PREMIA;
    }
#else

    /*////////////////////////////////////////*/
    static int cf_hes_varswap(double sigma0, double ka, double theta, double sigma
                                double Spot, double *fairval, double *Price)
    {
        double val, kk;
        double pvfactor = exp(-r * T);

// true values -----
        kk = ka * T;
        val = theta + (sigma0 - theta) * (1.0 - exp(-kk)) / kk;

        *fairval = val * 10000.0;

        *Price = pvfactor * (val * 10000.0 - Strike * Strike);
        return OK;
    }

    int CALC(CF_HES_VARIANCESWAP)(void *Opt, void *Mod, PricingMethod *Met)

```

```

{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;
    double r, divid, strike, spot;
    NumFunc_1 *p;

    r = log(1. + ptMod->R.Val.V_DOUBLE / 100.);
    divid = log(1. + ptMod->Divid.Val.V_DOUBLE / 100.);
    p = ptOpt->PayOff.Val.V_NUMFUNC_1;
    strike = p->Par[0].Val.V_DOUBLE;
    spot = ptMod->S0.Val.V_DOUBLE;

    return cf_hes_varswap(
        ptMod->Sigma0.Val.V_PDOUBLE
        , ptMod->MeanReversion.Val.V_PDOUBLE,
        ptMod->LongRunVariance.Val.V_PDOUBLE,
        ptMod->Sigma.Val.V_PDOUBLE,
        ptMod->Rho.Val.V_PDOUBLE,
        r, divid,
        ptOpt->Maturity.Val.V_DATE - ptMod->T.Val.V_DATE,
        strike, spot,
        &(Met->Res[0].Val.V_DOUBLE)/*FAIRVAL*/,
        &(Met->Res[1].Val.V_DOUBLE)/*PRICE*/);
}

static int CHK_OPT(CF_HES_VARIANCESWAP)(void *Opt, void *Mod)
{
    if ((strcmp(((Option *)Opt)->Name, "VarianceSwap") == 0))
        return OK;

    return WRONG;
}

#endif //PremiaCurrentVersion
static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    return OK;
}

```

```
PricingMethod MET(CF_HES_VARIANCESWAP) =
{
    "CF_HES_VARIANCESWAP",
    { {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CALC(CF_HES_VARIANCESWAP),
    { {"Fair strike for variance swap", DOUBLE, {100}, FORBID},
      {"Price in 10000 variance points", DOUBLE, {100}, FORBID},
      {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },
    CHK_OPT(CF_HES_VARIANCESWAP),
    CHK_ok ,
    MET(Init)
} ;

/*////////////////////////////////////////*/
}
```