

[Help](#)

```

#include "doublim.h"

static NumFunc_1 call =
{
    Call,
    {"Strike", PDOUBLE, {100}, ALLOW, SETABLE}, {" ", PREMIA_NULLTYPE, {0}, FORBI
    CHK_call
};

static NumFunc_1 const_Re =
{
    Const,
    {"Const Rebate", DOUBLE, {100}, ALLOW, SETABLE}, {" ", PREMIA_NULLTYPE, {0},
    CHK_ok
};

static NumFunc_1 const_Low =
{
    Const,
    {"Lower Limit", PDOUBLE, {100}, ALLOW, SETABLE}, {" ", PREMIA_NULLTYPE, {0},
    CHK_call
};

static NumFunc_1 const_Up =
{
    Const,
    {"Upper Limit", PDOUBLE, {100}, ALLOW, SETABLE}, {" ", PREMIA_NULLTYPE, {0},
    CHK_call
};

static TYPEOPT DoubleCallOutAmer =
{
    /*PayOff*/          {"PayOff", NUMFUNC_1, {0}, FORBID, SETABLE},
    /*Rebate*/          {"Const Rebate", NUMFUNC_1, {0}, FORBID, SETABLE},
    /*LowerLimit*/      {"Lower Limit", NUMFUNC_1, {0}, FORBID, SETABLE},
    /*UpperLimit*/      {"Upper Limit", NUMFUNC_1, {0}, FORBID, SETABLE},
    /*Maturity*/        {"Maturity", DATE, {0}, ALLOW, SETABLE},
    /*DateBetween0andMaturity*/ {"DateBetween0andMaturity", DATE, {0}, FORBID, UNSE
    /*OutOrIn*/         {"Out", BOOL, {OUT}, FORBID, UNSETABLE},

```

```

/*Parisian*/      {"Parisian", BOOL, {FALSE}, FORBID, UNSETABLE},
/*TwoDoubleStep*/ {"TwoDoubleStep", BOOL, {FALSE}, FORBID, UNSETABLE},
/*RebNo*/         {"Rebate", BOOL, {REBATE}, FORBID, UNSETABLE},
/*EuOrAm*/        {"Amer", BOOL, {AMER}, FORBID, UNSETABLE}

};

static int OPT(Init)(Option *opt, Model *mod)
{
    TYPEOPT *pt = (TYPEOPT *) (opt->TypeOpt);

    if (opt->init == 0)
    {
        opt->init = 1;
        opt->nvar = 11;
        opt->nvar_setable = 5;

        pt->PayOff.Val.V_NUMFUNC_1 = &call;
        pt->Rebate.Val.V_NUMFUNC_1 = &const_Re;
        pt->LowerLimit.Val.V_NUMFUNC_1 = &const_Low;
        pt->UpperLimit.Val.V_NUMFUNC_1 = &const_Up;

        (pt->EuOrAm).Val.V_BOOL = AMER;
        (pt->OutOrIn).Val.V_BOOL = OUT;
        (pt->RebOrNo).Val.V_BOOL = REBATE;
        (pt->Maturity).Val.V_DATE = 1.0;

        (pt->PayOff.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE = 100.0;
        (pt->Rebate.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE = 0.0;
        (pt->LowerLimit.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE = 90.0;
        (pt->UpperLimit.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE = 110.0;

        /* test for setability */
        if ((pt->RebOrNo).Val.V_BOOL == REBATE)
            pt->Rebate.Vsetable = SETABLE;
        else
            pt->Rebate.Vsetable = UNSETABLE;
    }

    return OK;
}

```

}

MAKEOPT(DoubleCallOutAmer);