

[Help](#)

```
#include "lim.h"
```

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

```
static NumFunc_1 rebate=
{
    Const,
    {"Rebate",PDOUBLE,{100},ALLOW,SETABLE},{ " ",PREMIA_NULLTYPE,{0},FORBID,SETABLE}},
    CHK_digit
};
```

```
static NumFunc_1 limit=
{
    ConstLim,
    {
        {"StartingDate",DATE,{0},IRRELEVANT,UNSETABLE},
        {"FinalDate",DATE,{0},IRRELEVANT,UNSETABLE},
        {"Frequency",PDOUBLE,{0},IRRELEVANT,UNSETABLE},
        {"Limit",PDOUBLE,{90},ALLOW,SETABLE},
        {"Delay",SPDOUBLE,{0},ALLOW,SETABLE},
        {" ",PREMIA_NULLTYPE,{0},FORBID,SETABLE}
    },
    CHK_digit
};
```

```
static TYPEOPT ParisianCallDownInEuro=
{
    /*Maturity*/      {"Maturity",DATE,{0},ALLOW,SETABLE},
    /*Limit*/         {"Limit",NUMFUNC_1,{0},FORBID,SETABLE},
    /*PayOff*/        {"Payoff",NUMFUNC_1,{0},FORBID,SETABLE}
    ,

    /*Rebate*/        {"Rebate",NUMFUNC_1,{0},FORBID,UNSETA
```

```

    BLE},
    /*OutOrIn*/      {"In",BOOL,{IN},FORBID,UNSETABLE},
    /*Parisian*/     {"Parisian",BOOL,{0},FORBID,UNSETABLE},

    /*DownOrUp*/     {"Down",BOOL,{DOWN},FORBID,UNSETABLE},
    /*RebNo*/        {"Rebate",BOOL,{REBATE},FORBID,UNSETABLE},
    BLE},
    /*EuOrAm*/       {"Euro",BOOL,{EURO},FORBID,UNSETABLE},
    /*PartOrTot*/    {"Total",BOOL,{TOTAL},FORBID,UNSETABLE}
    ,
    /*ContOrDisc*/   {"Cont",BOOL,{CONT},FORBID,UNSETABLE},
    /*ConstLim*/     {"ConstLim",BOOL,{CONSTLIM},ALLOW,SETABLE},
    BLE},
};

static int OPT(Init)(Option *opt,Model *mod)
{
    TYPEOPT* pt=( TYPEOPT*)(opt->TypeOpt);
    if ( opt->init == 0)
    {
        opt->init = 1;
        opt->nvar = 12;
        opt->nvar_setable = 3;

        pt->PayOff.Val.V_NUMFUNC_1=&call;
        pt->Rebate.Val.V_NUMFUNC_1=&rebate;
        pt->Limit.Val.V_NUMFUNC_1=&limit;

        (pt->PayOff.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE=10
0.0;
        (pt->Rebate.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE=10
.0;

        (pt->OutOrIn).Val.V_BOOL=IN;
        (pt->Parisian).Val.V_BOOL=OK;
        (pt->DownOrUp).Val.V_BOOL=DOWN;
        (pt->RebOrNo).Val.V_BOOL=NOREBATE;
        (pt->EuOrAm).Val.V_BOOL=EURO;
        (pt->PartOrTot).Val.V_BOOL=TOTAL;
        (pt->ContOrDisc).Val.V_BOOL=CONT;
    }
}

```

```
(pt->ConstLim).Val.V_BOOL=CONSTLIM;

(pt->Limit.Val.V_NUMFUNC_1)->Par[3].Val.V_PDOUBLE=90.
0;
(pt->Limit.Val.V_NUMFUNC_1)->Par[4].Val.V_SPDOUBLE=0.
01;
(pt->Maturity).Val.V_DATE=1.0;

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

}

return OK;
}

MAKEOPT(ParisianCallDownInEuro);
```

References