

[Help](#)

```

#include "lim.h"

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

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

static NumFunc_1 limit=
{
    ConstLim,
    {
        {"StartingDate",DATE,0,IRRELEVANT},
        {"FinalDate",DATE,0,IRRELEVANT},
        {"Frequency",PDOUBLE,0,IRRELEVANT},
        {"Limit",PDOUBLE,90,ALLOW},
        {" ",END,0,FORBID}
    },
    CHK_digit
};

static TYPEOPT CallUpOutParisianEuro=
{
    /*PayOff*/      {"Payoff",NUMFUNC_1,0,FORBID}
    ,
    /*Rebate*/      {"Rebate",NUMFUNC_1,0,FORBID}
    ,
    /*OutOrIn*/     {"Out",BOOL,OUT,FORBID},
    /*DownOrUp*/    {"Up",BOOL,UP,FORBID},
    /*Parisian*/    {"Parisian",BOOL

```

```
,0,FORBID},
/*RebNo*/      {"Rebate",BOOL,REBATE,FORBID}
,
/*EuOrAm*/      {"Euro",BOOL,EURO,FORBID},
/*PartOrTot*/   {"Total",BOOL,TOTAL,FORBID},
/*ContOrDisc*/  {"Cont",BOOL,CONT,FORBID},
/*ConstLim*/    {"ConstLim",BOOL,CONSTLIM,ALLOW},
/*Limit*/       {"Limit",NUMFUNC_1,0,FORBID},
/*Delay*/       {"Delay",PDOUBLE,DELAY,FORBID}
},
/*Maturity*/    {"Maturity",DATE,0,ALLOW}
};
```

```
static int OPT(Init)(Option *opt)
{
  TYPEOPT* pt=( TYPEOPT*)(opt->TypeOpt);
  static int first=1;

  if (first)
  {
    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=100.0;
    (pt->Rebate.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE=10.0;

    (pt->OutOrIn).Val.V_BOOL=OUT;
    (pt->DownOrUp).Val.V_BOOL=UP;
    (pt->RebOrNo).Val.V_BOOL=REBATE;
    (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=110.0;
```

```
        (pt->Maturity).Val.V_DATE=1.0;  
        (pt->Maturity).Val.V_DATE=0.2;  
  
        first=0;  
    }  
  
    return OK;  
}  
  
MAKEOPT(CallUpOutEuro);
```

References