

Help

```

#include "stda.h"
#include "error_msg.h"
#include "premia_obj.h"

static NumFunc_1 put =
{
    Put,
    {"Strike", PDOUBLE, {100}, FORBID, UNSETABLE}, {" ",
        PREMIA_NULLTYPE, {0}, FORBID, SETABLE}},
    CHK_call
};

static TYPEOPT GMWB =
{
    /*PayOff*/ {"Payoff", NUMFUNC_1, {0}, FORBID, UNSETABLE},
    /*EuOrAm*/ {"Euro", BOOL, {AMER}, FORBID, UNSETABLE},
    /*Maturity*/ {"Maturity", DATE, {0}, ALLOW, SETABLE},
    /*DeemedContribution*/ {"Deemed Contribution", PDOUBLE,
        {0}, IRRELEVANT, UNSETABLE},
    /*InitialAge*/ {"Initial Age", PDOUBLE, {0}, FORBID, UNSETABLE},
    /*Premium*/ {"Premium", PDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /*MinimumGuaranteed*/ {"MinimumGuaranteed", PDOUBLE, {0},
        IRRELEVANT, UNSETABLE},
    /*Number of Monitoring Dates*/ {"Number of Monitoring
        Dates", PINT, {0}, ALLOW, SETABLE},
    /*Alpha*/ {"Alpha", RGDOUBLE, {0}, FORBID, UNSETABLE},
    /*Alpha_m*/ {"Alpha_m", RGDOUBLE, {0}, ALLOW, SETABLE},
    /*MultiplierCPPi*/ {"MultiplierCPPi", PDOUBLE, {0}, IR
        RELEVANT, UNSETABLE},
    /*Ratchet*/ {"Ratchet at the Monitoring Dates(Boolean)",
        BOOL, {0}, FORBID, UNSETABLE},
    /*Gamma*/ {"Gamma", PDOUBLE, {0}, FORBID, UNSETABLE},
    /*Bonus B*/ {"Bonus", PDOUBLE, {0}, FORBID, UNSETABLE},
    /*WithdrawlRate G*/ {"WithdrawlRate", PDOUBLE, {0}, FORB
        ID, UNSETABLE},
    /*Base case surrender charges*/ {"SurrenderCharges", PNLV

```

```

    ECT, {0}, FORBID, SETABLE},
/*Base case surrender Times*/{"SurrenderTimes", PNLVECT,
    {0}, FORBID, SETABLE},
/*Mortality*/{"MortalityData", FILENAME, {0}, FORBID, UN
    SETABLE},
/*Maximum WithdrawalRate G*/ {"MaximumWithdrawalRate",
    PDOUBLE, {0}, ALLOW, SETABLE},

/*Rateaccumulation*/ {"RateAccumulation", PDOUBLE, {0}, FO
    RBID, UNSETABLE},
/*PremiumPercentage*/ {"PremiumPercentage", PDOUBLE, {0
    }, FORBID, UNSETABLE},
/*RollUpRate*/ {"CompoundRollUpRate", PDOUBLE, {0}, FORB
    ID, UNSETABLE},
/*ForceOfMortality*/ {"ForceOfMortality", PDOUBLE, {0},
    FORBID, UNSETABLE},
/*TermCertainAnnuitiyMaturity*/ {"TermCertainAnnuitiyMatu
    rity", DATE, {0},FORBID,UNSETABLE},
};

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

    if (opt->init == 0)
    {
        opt->init = 1;
        opt->nvar = 24;
        opt->nvar_setable = 6;
        pt->PayOff.Val.V_NUMFUNC_1 = &put;
        pt->SurrenderCharges.Val.V_PNLVECT = NULL;
        pt->SurrenderTimes.Val.V_PNLVECT = NULL;

        (pt->Maturity).Val.V_DATE = 10;
        (pt->MaximumWithdrawalRate).Val.V_PDOUBLE = 10;
        (pt->NumberofMonitoringDates).Val.V_PINT = 10;
        (pt->Alpha_m).Val.V_RGDOUBLE = 0.01;

        /* SurrenderCharges */
        if ((pt->SurrenderCharges).Val.V_PNLVECT == NULL)

```

```
    {
        double SurrenderCharges[7] = {0.08, 0.07, 0.06, 0
.05, 0.04, 0.03,0.};

        if ((pt->SurrenderCharges.Val.V_PNLVECT =
            pnl_vect_create_from_ptr(7, SurrenderCharg
es)) == NULL)
            return WRONG;
    }

    /* SurrenderTimes */
    if ((pt->SurrenderTimes).Val.V_PNLVECT == NULL)
    {
        double SurrenderTimes[7] = {1.999, 2.999, 3.999,
4.999, 5.999, 6.999,100.};
        if ((pt->SurrenderTimes.Val.V_PNLVECT =
            pnl_vect_create_from_ptr(7, SurrenderTime
s)) == NULL)
            return WRONG;
    }

}

return OK;
}

MAKEOPT(GMWB);
```

References