

[Help](#)

```
#include "
href../..mod/bsld/bsld_stda/bsld_stda_h_src.pdfstda.h"
#include "
href../..common/error_msg_h_src.pdferror_msg.h"
#include " premia_obj.h"

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

static TYPEOPT EquityLinkedSurrenderEndowment =
{
    /* PayOff; */ {"Payoff", NUMFUNC_1, {0}, FORBID, UNSETABLE},
    /* EuOrAm */ {"Amer", BOOL, {AMER}, FORBID, UNSETABLE},
    /* Maturity */ {"Maturity(in years)", DATE, {0}, ALLOW, SETABLE},
    /* DeemedContribution */ {"Deemed Contribution", PDOUBLE, {0}, ALLOW, SETABLE},
    /* InitialAge */ {"Initial Age", PDOUBLE, {0}, ALLOW, SETABLE},
    /* Premium */ {"Premium", PDOUBLE, {0}, ALLOW, SETABLE},
    /* MinimumGuaranteed */ {"MinimumGuaranteed", PDOUBLE, {0}, ALLOW, SETABLE},
    /* Number of Monitoring Dates */ {"Number of Monitoring Dates", PINT, {0}, IRRELEVANT, UNSETABLE},
    /* Alpha */ {"Alpha", RGDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* Alpha_m */ {"Alpha_m", RGDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* MultiplierCPi */ {"MultiplierCPi", PDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* Ratchet */ {"Ratchet", BOOL, {0}, IRRELEVANT, UNSETABLE},
    /* Gamma */ {"Gamma", PDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* Bonus B */ {"Bonus", PDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* WithdrawlRate G */ {"WithdrawlRate", PDOUBLE, {0}, IRRELEVANT, UNSETABLE},
    /* Base case surrender charges */ {"SurrenderCharges", PNLVECT, {0}, IRRELEVANT, UNSETABLE},
    /* Base case surrender Times */ {"SurrenderTimes", PNLVECT, {0}, IRRELEVANT, UNSETABLE},
    /* Mortality */ {"MortalityData", FILENAME, {0}, FORBID, SETABLE},
    /* Maximum WithdrawlRate G */ {"Maximum WithdrawalRate", PDOUBLE, {0}, FORBID, UNSETABLE},
    /* RateAccumulation */ {"RateAccumulation", PDOUBLE, {0}, FORBID, UNSETABLE},
    /* PremiumPercentage */ {"PremiumPercentage", PDOUBLE, {0}, FORBID, UNSETABLE},
    /* RollUpRate */ {"CompoundRollUpRate", PDOUBLE, {0}, FORBID, UNSETABLE},
    /* ForceOfMortality */ {"ForceOfMortality", PDOUBLE, {0}, FORBID, UNSETABLE},
    /* TermCertainAnnuityMaturity */ {"TermCertainAnnuityMaturity", 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->HelpFilenameHint = "equity_linked";
        opt->nvar = 24;
        opt->nvar_setable = 7;

        pt->PayOff.Val.V_NUMFUNC_1 = &put;
        (pt->PayOff.Val.V_NUMFUNC_1)->Par[0].Val.V_PDOUBLE = 100.0;
        (pt->EuOrAm).Val.V_BOOL = AMER;

        (pt->Maturity).Val.V_DATE = 5.;
        (pt->Premium).Val.V_PDOUBLE = 106;
        (pt->DeemedContribution).Val.V_PDOUBLE = 100;
        (pt->MinimumGuaranteed).Val.V_PDOUBLE = 0.02;
        (pt->InitialAge).Val.V_PDOUBLE = 50;

        /*Mortality Data*/
        if (pt->MortalityData.Val.V_FILENAME == NULL)
        {
            if ((pt->MortalityData.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH))
                return MEMORY_ALLOCATION_FAILURE;
            sprintf(pt->MortalityData.Val.V_FILENAME, "%s%sMortality.dat", premia_
        }

    }
    return OK;
}

MAKEOPT(EquityLinkedSurrenderEndowment);

```