```
   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 GLWB =
{
  /*PayOff*/      {"Payoff", NUMFUNC_1, {0}, FORBID, UNSETA
    BLE},
  /*EuOrAm*/       {"Euro", BOOL, {AMER}, FORBID, UNSETABLE
    },
  /*Maturity*/     {"Maturity", DATE, {0}, ALLOW, SETABLE},
  /*DeemedContribution;*/ {"Deemed Contribution", PDOUBLE,
    {0}, IRRELEVANT, UNSETABLE},
  /*InitialAge*/ {"Initial Age", PDOUBLE, {0}, ALLOW, SETA
    BLE},
  /*Premium;*/ {"Premium", PDOUBLE, {0}, IRRELEVANT, UNSETA
    BLE},
  /*MinimumGuaranteed*/ {"MinimumGuaranteed", PDOUBLE, {0},
     IRRELEVANT, UNSETABLE},
  /*Number of Monitoring Dates*/ {"Number of Monitoring Da
    tes", PINT, {0}, IRRELEVANT, UNSETABLE},
  /*Alpha*/ {"Alpha", RGDOUBLE, {0}, FORBID, UNSETABLE},
  /*Alpha_m*/ {"Alpha_m", RGDOUBLE, {0}, ALLOW, SETABLE},
  /*MultiplieriCPPi*/ {"MultiplieriCPPi", PDOUBLE, {0}, IR
    RELEVANT, UNSETABLE},
  /*Ratchet*/{"Ratchet at the Monitoring Dates(Boolean)",
    BOOL, {0}, ALLOW, SETABLE},
  /*Gamma*/ {"Gamma", PDOUBLE, {0}, ALLOW, SETABLE},
  /*Bonus B*/ {"Bonus", PDOUBLE, {0}, ALLOW, SETABLE},
  /*WithdrawalRate G*/ {"WithdrawalRate", PDOUBLE, {0}, ALL
    OW, SETABLE},
  /*Base case surrender charges.*/{"SurrenderCharges", PNLV
```

```
    ECT, {0}, FORBID, SETABLE},
    /*Base case surrender Times.*/{"SurrenderTimes", PNLVEC
    T, {0}, FORBID,SETABLE},

  /*Mortality*/{"MortalityData", FILENAME, {0}, FORBID, SE
    TABLE},
  /*Maximum WithdrawlRate G*/ {"MaximumWithdrawlRate", PDO
    UBLE, {0}, FORBID, UNSETABLE},
  /*RateAccumulation*/ {"RateAccumulation", PDOUBLE, {0},
    FORBID, UNSETABLE},
  /*PremiumPercentuage*/ {"PremiumPercentuage", PDOUBLE, {0
    }, FORBID, UNSETABLE},
  /*RollUpRate*/ {"CompoundRollUpRate", PDOUBLE, {0}, FORB
    ID, UNSETABLE},
/*ForceOfMortality*/ {"ForceOfMortality", PDOUBLE, {0}, FO
    RBID, 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 = 9;

      (pt->Maturity).Val.V_DATE = 56;
      (pt->Alpha_m).Val.V_RGDOUBLE = 0.0;
      (pt->InitialAge).Val.V_PDOUBLE = 65;
      (pt->Ratchet).Val.V_BOOL = 0;
      (pt->Gamma).Val.V_PDOUBLE = 1;
      (pt->WithdrawalRate).Val.V_PDOUBLE = 0.05;
      (pt->Bonus).Val.V_PDOUBLE = 0.05;
      pt->MortalityData.Val.V_FILENAME = NULL;
      pt->SurrenderCharges.Val.V_PNLVECT = NULL;
    pt->SurrenderTimes.Val.V_PNLVECT = NULL;
```

```
      pt->PayOff.Val.V_NUMFUNC_1 = &put;

    if ((pt->SurrenderCharges).Val.V_PNLVECT == NULL)
        {
          double SurrenderCharges[6] = {0.05, 0.04, 0.03, 0
    .02, 0.01, 0.0};
          if ((pt->SurrenderCharges.Val.V_PNLVECT =
                 pnl_vect_create_from_ptr(6, SurrenderCharg
    es)) == NULL)
            return WRONG;
        }


     /* SurrenderTimes */
      if ((pt->SurrenderTimes).Val.V_PNLVECT == NULL)
        {
          double SurrenderTimes[6] = {1., 2., 3., 4., 5., 6
    .};
          if ((pt->SurrenderTimes.Val.V_PNLVECT =
                 pnl_vect_create_from_ptr(6, SurrenderTime
    s)) == NULL)
            return WRONG;
        }

      /*Mortality Data*/
      if (pt->MortalityData.Val.V_FILENAME == NULL)
        {
          if ((pt->MortalityData.Val.V_FILENAME = malloc(si
    zeof(char) * MAX_PATH_LEN)) == NULL)
            return MEMORY_ALLOCATION_FAILURE;
          sprintf(pt->MortalityData.Val.V_FILENAME, "%s%sm
    ortalityDAV2004R.dat", premia_data_dir, path_sep);
        }
    }

  return OK;
}

MAKEOPT(GLWB);
```

References