```
   Help
#include "stdndc.h"
#include "error_msg.h"
#include "premia_obj.h"

static TYPEOPT CDO_COPULA =
{
    {"Number of Companies", PINT, {0}, FORBID, UNSETABLE},
    {"Maturity",DATE,{0},ALLOW,SETABLE},
    {"Homogeneous Nominals", ENUM, {0}, FORBID, SETABLE},
    {"Tranches", PNLVECT, {0}, FORBID, SETABLE},
    {"Type of Recovery",ENUM,{0},FORBID,SETABLE},
    {"Recovery Parameters",DOUBLE,{0},IRRELEVANT,UNSETABLE}
    ,
    {"Number of coupon payments per year",INT,{0},ALLOW,SE
    TABLE},
    {"Current date", DATE, {0}, IRRELEVANT, UNSETABLE},
    {"Number of defaults at current date", INT, {0}, IRREL
    EVANT, UNSETABLE}

};

static PremiaEnumMember RecoveryTypeMembers[] =
{
    { "Constant", 1, 1 },
    { "Uniform", 2, 1 },
    { NULL, NULLINT, 0 }
};

static PremiaEnumMember NominalTypeMembers[] =
{
    { "Homogeneous", 1, 0 },
    { "Non homogeneous", 2, 1 },
    { NULL, NULLINT, 0 }
};

static DEFINE_ENUM(RecoveryType, RecoveryTypeMembers);
static DEFINE_ENUM(NominalType, NominalTypeMembers);

static int OPT(Init)(Option *opt,Model *mod)
{
```

```
double   t[2];
VAR     *Par;
TYPEOPT *pt          = (TYPEOPT*)(opt->TypeOpt);
VAR     *ptMod       = (VAR*)(mod->TypeModel);
int      n_recovery = 0;

/* get the size from the model */
mod->Init(mod);
pt->Ncomp.Val.V_PINT = ptMod[0].Val.V_PINT;

if (opt->init == 0 )
  {
    opt->init = 1;
    opt->nvar = 8;

    pt->maturity.Val.V_DATE=5.0;
    pt->date.Val.V_DATE = 0.; /* useless but needs to be
  initialiased */
    pt->n_defaults.Val.V_INT = 0; /* useless but needs
  to be initialiased */
    pt->NbPayment.Val.V_INT=4;


    opt->nvar_setable = 6;

    pt->t_nominal.Viter = FORBID;
    pt->t_nominal.Vsetable = SETABLE;
    pt->t_recovery.Viter = FORBID;
    pt->t_recovery.Vsetable = SETABLE;


    pt->t_nominal.Val.V_ENUM.value=1;
    pt->t_nominal.Val.V_ENUM.members=&NominalType;
    Par = lookup_premia_enum_par (&(pt->t_nominal), 2);
    Par[0].Viter=FORBID;
    Par[0].Vsetable = SETABLE;
    Par[0].Vtype = FILENAME;
    Par[0].Val.V_FILENAME = NULL;
    Par[0].Vname = "Nominal data";
```

```
    pt->tranch.Val.V_PNLVECT=NULL;

    pt->t_recovery.Val.V_ENUM.value = 1;
    pt->t_recovery.Val.V_ENUM.members = &RecoveryType;

    Par = lookup_premia_enum_par (&(pt->t_recovery), 1);
    Par[0].Viter=FORBID;
    Par[0].Vsetable = SETABLE;
    Par[0].Vtype = DOUBLE;
    Par[0].Val.V_DOUBLE=0.4;
    Par[0].Vname = "Recovery";

    Par = lookup_premia_enum_par (&(pt->t_recovery), 2);
    Par[0].Viter=FORBID;
    Par[0].Vsetable = SETABLE;
    Par[0].Vtype = PNLVECT;
    Par[0].Val.V_PNLVECT=NULL;
    Par[0].Vname = "Recovery";
  }

/* Recovery vector */
    Par = lookup_premia_enum_par (&(pt->t_recovery), 2);
    if ( Par[0].Val.V_PNLVECT == NULL )
      {
        if (n_param_recovery (&n_recovery, t,  2, 1) !=
  OK) return WRONG;
      Par[0].Val.V_PNLVECT=pnl_vect_create_from_ptr (n_
  recovery, t);
    }


/* tranches */
if ((pt->tranch).Val.V_PNLVECT == NULL)
  {
    double tranches[5] = {0, 0.03, 0.06, 0.1, 1};
    if ((pt->tranch.Val.V_PNLVECT =
         pnl_vect_create_from_ptr (5, tranches))==NULL)
      return WRONG;
  }

/* Nominal filename */
```

```
  Par = lookup_premia_enum_par (&(pt->t_nominal), 2);
  if ( Par[0].Val.V_FILENAME == NULL )
    {
      if ((Par[0].Val.V_FILENAME= malloc(sizeof(char)*MAX_
    PATH_LEN))==NULL)
        return MEMORY_ALLOCATION_FAILURE;
      sprintf(Par[0].Val.V_FILENAME, "%s%scdo_nominal.dat",
     premia_data_dir, path_sep);
    }
  return OK;
}

MAKEOPTGEN(CDO_COPULA);
```

References