

[Help](#)

```
#include "
href../../mod/dynamic/dynamic_stdndc/dynamic_stdndc_h_src.pdfstdndc.h"
#include "
href../../common/error_msg_h_src.pdferror_msg.h"
#include "premia_obj.h"

static TYPEOPT CDO =
{
    {"Number of Companies", PINT, {0}, FORBID, UNSETABLE},
    {"Maturity", DATE, {0}, ALLOW, SETABLE},
    {"Homogeneous Nominals", ENUM, {0}, FORBID, SETABLE},
    {"Tranches", PNLVECT, {0}, FORBID, SETABLE},
    {"Number of coupon payments per year", INT, {0}, ALLOW, SETABLE},
    {"Current date", DATE, {0}, IRRELEVANT, UNSETABLE},
    {"Number of defaults at current date", INT, {0}, IRRELEVANT, UNSETABLE}
};

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

static DEFINE_ENUM(NominalType, NominalTypeMembers);

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

    /* 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 = 7;

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

    opt->nvar_setable = 4;

    pt->t_nominal.Viter = FORBID;
    pt->t_nominal.Vsetable = SETABLE;
    if (strcmp(mod->Name, "Hawkes_Intensity") == 0 ||
        strcmp(mod->Name, "Dynamic") == 0 ||
        strcmp(mod->Name, "SLI"))
        pt->t_nominal.Vsetable = UNSETABLE;
    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;
}

/* 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%scto_nominal.dat", premia_data_dir, pat
}
return OK;
}

MAKEOPT(CDO);

```