

[Help](#)

```
#include "
href../../mod/black_cox_extended/black_cox_extended_h_src.pdfblack_cox_extended.
#include "
href../../common/chk_h_src.pdfchk.h"
#include "
href../../common/error_msg_h_src.pdferror_msg.h"
#include "
href../../mod/hes1d/hes1d_pad/model_h_src.pdfmodel.h"

static int MOD(Init)(Model *model)
{
    TYPEMOD *pt = (TYPEMOD *)(model->TypeModel);
    int n = 2;

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

        pt->S0.Vname = "Spot";
        pt->S0.Vtype = PDOUBLE;
        pt->S0.Val.V_PDOUBLE = 100.;
        pt->S0.Viter = ALLOW;
        model->nvar++;

        pt->R.Vname = "Annual Interest Rate";
        pt->R.Vtype = DOUBLE;
        pt->R.Val.V_DOUBLE = 10.;
        pt->R.Viter = ALLOW;
        model->nvar++;

        pt->Sigma.Vname = "Volatility";
        pt->Sigma.Vtype = DOUBLE;
        pt->Sigma.Val.V_DOUBLE = 0.2;
        pt->Sigma.Viter = ALLOW;
        model->nvar++;

        pt->L.Vname = "Barrier";
        pt->L.Vtype = PNLVECT;
```

```

    pt->L.Val.V_PNLVECT = NULL;
    pt->L.Viter = FORBID;
    model->nvar++;

    pt->alpha.Vname = "Barrier decrease";
    pt->alpha.Vtype = DOUBLE;
    pt->alpha.Val.V_DOUBLE = 0.1;
    pt->alpha.Viter = FORBID;
    model->nvar++;

    pt->mu.Vname = "Default intensities";
    pt->mu.Vtype = PNLVECT;
    pt->mu.Val.V_PNLVECT = NULL;
    pt->mu.Viter = FORBID;
    model->nvar++;
}

if (pt->L.Val.V_PNLVECT == NULL)
{
    if ((pt->L.Val.V_PNLVECT = pnl_vect_create_from_list(n, 95., 85.)) == NULL)
        goto err;
}
if (pt->mu.Val.V_PNLVECT == NULL)
{
    if ((pt->mu.Val.V_PNLVECT =
        pnl_vect_create_from_list(n + 1, 0.05, 0.2, 0.3)) == NULL)
        goto err;
}
return OK;

err:
    Fprintf(TOSCREEN, "%s\ n", error_msg[MEMORY_ALLOCATION_FAILURE]);
    exit(WRONG);
}

TYPEMOD black_cox_extended;
MAKEMOD(black_cox_extended);

```