

[Help](#)

```
#include "
href../../mod/hesvasicek2d/hesvasicek2d_h_src.pdfhesvasicek2d.h"
#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"
#include "pnl/pnl_vector.h"

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

    static double vrq[] = {0.2, 0.04, 0.03};
    static double kappa[] = {2, 1, 1};
    static double theta[] = {0.2, 0.04, 0.03};
    static double sigma[] = {0.3, 0.3, 0.3};
    static double rho[] = {0, 0, 0, 0., 0., 0.};

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

        pt->T.Vname = "Current Date";
        pt->T.Vtype = DATE;
        pt->T.Val.V_DATE = 0.0;
        pt->T.Viter = ALLOW;
        model->nvar++;

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

        pt->vrq.Vname = "V0 r0 divid0";
        pt->vrq.Vtype = PNLVECT;
```

```

pt->vrq.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, vrq);
pt->vrq.Viter = FORBID;
model->nvar++;

pt->kappa.Vname = "Speed k : V r divid";
pt->kappa.Vtype = PNLVECT;
pt->kappa.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, kappa);
pt->kappa.Viter = FORBID;
model->nvar++;

pt->theta.Vname = " theta : V r divid";
pt->theta.Vtype = PNLVECT;
pt->theta.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, theta);
pt->theta.Viter = FORBID;
model->nvar++;

pt->sigma.Vname = "Sigma : V r divid";
pt->sigma.Vtype = PNLVECT;
pt->sigma.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, sigma);
pt->sigma.Viter = FORBID;
model->nvar++;

pt->rho.Vname = "rhoSv rhoSr rhoSdivid rhovr rhovdivid rhordivid";
pt->rho.Vtype = PNLVECT;
pt->rho.Val.V_PNLVECT = pnl_vect_create_from_ptr(6, rho);
pt->rho.Viter = FORBID;
model->nvar++;
}

if (pt->sigma.Val.V_PNLVECT == NULL)
{
    if ((pt->sigma.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.03)) == NULL)
        goto err;
}

if (pt->rho.Val.V_PNLVECT == NULL)
{
    if ((pt->rho.Val.V_PNLVECT = pnl_vect_create_from_double(6, 0.0)) == NULL)
        goto err;
}

```

```

    }

    if (pt->theta.Val.V_PNLVECT == NULL)
    {
        if ((pt->theta.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.)) == NULL)
            goto err;
    }

    if (pt->vrq.Val.V_PNLVECT == NULL)
    {
        if ((pt->vrq.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.05)) == NULL)
            goto err;
    }

    if (pt->vrq.Val.V_PNLVECT == NULL)
    {
        if ((pt->vrq.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.05)) == NULL)
            goto err;
    }

    if (pt->kappa.Val.V_PNLVECT == NULL)
    {
        if ((pt->kappa.Val.V_PNLVECT = pnl_vect_create_from_double(3, 2.)) == NULL)
            goto err;
    }

    return OK;

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

TYPEMOD hesvasicek2d;
MAKEMOD(hesvasicek2d);

```