

[Help](#)

```
#include "hbw4d.h"
#include "chk.h"
#include "error_msg.h"
#include "model.h"
#include "pnl/pnl_vector.h"

extern char *path_sep;

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

    static double x0[] = {1.35, 0.1};
    static double r[] = {0.02, 0.05};
    static double kappa[] = {0.5, 0.01, 0.05};
    static double sigma[] = {0.3, 0.007, 0.012};
    static double rho[] = { -0.4, -0.15, -0.15, 0.3, 0.3, 0.25};

    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->r.Vname = "Interest rates : rd(rn) rf(rr)";
        pt->r.Vtype = PNLVECT;
        pt->r.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, r);
        pt->r.Viter = FORBID;
        model->nvar++;

        pt->MeanReversion.Vname = "Mean Reversion";
        pt->MeanReversion.Vtype = DOUBLE;
        pt->MeanReversion.Val.V_DOUBLE = 0.1;
```

```

pt->MeanReversion.Viter = ALLOW;
model->nvar++;

pt->x0.Vname = "Initial : csi0(IO) v0";
pt->x0.Vtype = PNLVECT;
pt->x0.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, x0);
pt->x0.Viter = FORBID;
model->nvar++;

pt->kappa.Vname = "Speed : k lambdad(an) lambdaf(ar)";
pt->kappa.Vtype = PNLVECT;
pt->kappa.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, kappa);
pt->kappa.Viter = FORBID;
model->nvar++;

pt->sigma.Vname = "Sigma : gamma(sigma_v) etad(etan) etaf(etar)";
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 = "rho12 rho13 rho14 rho23 rho24 rho34";
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, 0.02)) == NULL)
        goto err;
}

if (pt->rho.Val.V_PNLVECT == NULL)
{
    if ((pt->rho.Val.V_PNLVECT = pnl_vect_create_from_double(6, 0.02)) == NULL)
        goto err;
}

```

```
if (pt->x0.Val.V_PNLVECT == NULL)
{
    if ((pt->x0.Val.V_PNLVECT = pnl_vect_create_from_double(2, 0.)) == NULL)
        goto err;
}

if (pt->r.Val.V_PNLVECT == NULL)
{
    if ((pt->r.Val.V_PNLVECT = pnl_vect_create_from_double(2, 0.)) == NULL)
        goto err;
}

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

return OK;

err:
    Fprintf(TOSCREEN, "%s\ n", error_msg[MEMORY_ALLOCATION_FAILURE]);
    exit(WRONG);
}
TYPEMOD HHW4d;
MAKEMOD(HHW4d);
```