

[Help](#)

```
#include "wishart2d.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 S0[] = {100., 100.};
    static double Divid[] = {0., 0.};
    static double b[] = {0., 0.};
    static double Sigma0[] = {0.02, 0.02, 0.02, 0.03};
    static double Q[] = {0.02, 0.02, 0.02, 0.03};

    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 = "Initial Spot";
        pt->S0.Vtype = PNLVECT;
        pt->S0.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, S0);
        pt->S0.Viter = FORBID;
        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->Divid.Vname = "Annual Dividend Rate";
pt->Divid.Vtype = PNLVECT;
pt->Divid.Val.V_PNLVECT = pnl_vect_create_from_ptr(2, Divid);
pt->Divid.Viter = FORBID;
model->nvar++;

pt->alpha.Vname = "alpha";
pt->alpha.Vtype = DOUBLE;
pt->alpha.Val.V_DOUBLE = 3.;
pt->alpha.Viter = ALLOW;
model->nvar++;

pt->b.Vname = "b-11 b-12 b-21 b-22";
pt->b.Vtype = PNLVECT;
pt->b.Val.V_PNLVECT = pnl_vect_create_from_ptr(4, b);
pt->b.Viter = FORBID;
model->nvar++;

pt->Sigma0.Vname = "X0-11 X0-12 X0-21 X0-22";
pt->Sigma0.Vtype = PNLVECT;
pt->Sigma0.Val.V_PNLVECT = pnl_vect_create_from_ptr(4, Sigma0);
pt->Sigma0.Viter = FORBID;
model->nvar++;

pt->Q.Vname = "Q-11 Q-12 Q-21 Q-22";
pt->Q.Vtype = PNLVECT;
pt->Q.Val.V_PNLVECT = pnl_vect_create_from_ptr(4, Q);
pt->Q.Viter = FORBID;
model->nvar++;

}
if (pt->S0.Val.V_PNLVECT == NULL)
{
    if ((pt->S0.Val.V_PNLVECT = pnl_vect_create_from_double(2, 100.)) == NULL)
        goto err;
}

if (pt->Divid.Val.V_PNLVECT == NULL)
{

```

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

    if (pt->Sigma0.Val.V_PNLVECT == NULL)
    {
        if ((pt->Sigma0.Val.V_PNLVECT = pnl_vect_create_from_double(4, 0.02)) == NULL)
            goto err;
    }

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

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

    return OK;

err:
    Fprintf(TOSCREEN, "%s\ n", error_msg[MEMORY_ALLOCATION_FAILURE]);
    exit(WRONG);
}
TYPEMOD WISHART2d;
MAKEMOD(WISHART2d);
```