Help

```
#include "varswap3d.h"
#include "chk.h"
#include "error_msg.h"
#include "model.h"
#include "pnl/pnl_matrix.h"

extern char *path_sep;

static int MOD(Init)(Model *model)
{
  TYPEMOD *pt = (TYPEMOD *)(model->TypeModel);
  double Beta[] = {0.8, 0.5, 0.3};
  double MeanReversion[] = {2.0, 1.5, 0.8};

  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.;
      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->Divid.Vname = "Annual Dividend Rate";
      pt->Divid.Vtype = DOUBLE;
      pt->Divid.Val.V_DOUBLE = 0.;
      pt->Divid.Viter = ALLOW;
      model->nvar++;

      pt->R.Vname = "Annual Interest Rate";
      pt->R.Vtype = DOUBLE;
```

```
    pt->R.Val.V_DOUBLE = 0.0;
    pt->R.Viter = ALLOW;
    model->nvar++;

    pt->V0.Vname = "Current Variance";
    pt->V0.Vtype = DOUBLE;
    pt->V0.Val.V_DOUBLE = 0.2;
    pt->V0.Viter = ALLOW;
    model->nvar++;

    pt->Beta.Vname = "Volatility of Volatility";
    pt->Beta.Vtype = PNLVECT;
    pt->Beta.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, Beta);
    pt->Beta.Viter = FORBID;
    model->nvar++;

    pt->MeanReversion.Vname = "Mean Reversion Factor";
    pt->MeanReversion.Vtype = PNLVECT;
    pt->MeanReversion.Val.V_PNLVECT = pnl_vect_create_from_ptr(3, MeanReversio
    pt->MeanReversion.Viter = FORBID;
    model->nvar++;

    pt->Rho.Vname = "Correlation";
    pt->Rho.Vtype = RGDOUBLEM11;
    pt->Rho.Val.V_RGDOUBLEM11 = 0.;
    pt->Rho.Viter = ALLOW;
    model->nvar++;
  }
if (pt->Beta.Val.V_PNLVECT == NULL)
  {
    if ((pt->Beta.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.2)) == NULL
      goto err;
  }
if (pt->MeanReversion.Val.V_PNLVECT == NULL)
  {
    if ((pt->MeanReversion.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.2)
      goto err;
  }


return OK;
```

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

}
TYPEMOD VarSwap3dim;
MAKEMOD(VarSwap3dim);
```