Help

```
#include "
href../../mod/heshw2d/heshw2d_h_src.pdfheshw2d.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"
#include "premia_obj.h"
#include "
href../../common/enums_h_src.pdfenums.h"

double MOD(GetYield_rd)(TYPEMOD *pt)
{
  VAR *Par;
  Par = lookup_premia_enum_par(&(pt->flat_flag_rd), 0);
  return Par[0].Val.V_PDOUBLE;
}

char *MOD(GetCurve_rd)(TYPEMOD *pt)
{
  VAR *Par;
  Par = lookup_premia_enum_par(&(pt->flat_flag_rd), 1);
  return Par[0].Val.V_FILENAME;
}

double MOD(GetYield_rf)(TYPEMOD *pt)
{
  VAR *Par;
  Par = lookup_premia_enum_par(&(pt->flat_flag_rf), 0);
  return Par[0].Val.V_PDOUBLE;
}

char *MOD(GetCurve_rf)(TYPEMOD *pt)
{
  VAR *Par;
  Par = lookup_premia_enum_par(&(pt->flat_flag_rf), 1);
  return Par[0].Val.V_FILENAME;
```

```
}

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

  static double kappa[] = {2, 1, 1};
  static double sigma[] = {0.3, 0.2, 0.2};
  static double rho[] = {-0.5,-0.5, -0.5, 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->v0.Vname = "v0";
      pt->v0.Vtype = PDOUBLE;
      pt->v0.Val.V_PDOUBLE = 0.1;
      pt->v0.Viter = ALLOW;
      model->nvar++;

      pt->flat_flag_rd.Vname = "Initial Yield Curve RD";
      pt->flat_flag_rd.Vtype = ENUM;
      pt->flat_flag_rd.Val.V_ENUM.value = 0;
      pt->flat_flag_rd.Val.V_ENUM.members = &PremiaEnumFlat;
      pt->flat_flag_rd.Viter = ALLOW;
      model->nvar++;
      Par_rd = lookup_premia_enum_par(&(pt->flat_flag_rd), 0);
```

```
Par_rd[0].Vname = "Yield Value RD";
Par_rd[0].Vtype = PDOUBLE;
Par_rd[0].Val.V_PDOUBLE = 0.04;
Par_rd[0].Viter = ALLOW;
Par_rd = lookup_premia_enum_par(&(pt->flat_flag_rd), 1);
Par_rd[0].Vname = "Yield Curve RD";
Par_rd[0].Vtype = FILENAME;
Par_rd[0].Val.V_FILENAME = NULL;
Par_rd[0].Viter = FORBID;

pt->flat_flag_rf.Vname = "Initial Yield Curve RF";
pt->flat_flag_rf.Vtype = ENUM;
pt->flat_flag_rf.Val.V_ENUM.value = 0;
pt->flat_flag_rf.Val.V_ENUM.members = &PremiaEnumFlatCopy;
pt->flat_flag_rf.Viter = ALLOW;
model->nvar++;
Par_rf = lookup_premia_enum_par(&(pt->flat_flag_rf), 0);
Par_rf[0].Vname = "Yield Value RF";
Par_rf[0].Vtype = PDOUBLE;
Par_rf[0].Val.V_PDOUBLE = 0.03;
Par_rf[0].Viter = ALLOW;
Par_rf = lookup_premia_enum_par(&(pt->flat_flag_rf), 1);
Par_rf[0].Vname = "Yield Curve RF";
Par_rf[0].Vtype = FILENAME;
Par_rf[0].Val.V_FILENAME = NULL;
Par_rf[0].Viter = FORBID;

pt->kappa.Vname = "Speed k : V r rf";
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 = "Long-Run Variance theta";
pt->theta.Vtype = PDOUBLE;
pt->theta.Val.V_PDOUBLE = 0.1;
pt->theta.Viter = ALLOW;
model->nvar++;

pt->sigma.Vname = "Sigma : V r rf";
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 rhoSrf rhovr rhovrf rhorrf";
    pt->rho.Vtype = PNLVECT;
    pt->rho.Val.V_PNLVECT = pnl_vect_create_from_ptr(6, rho);
    pt->rho.Viter = FORBID;
    model->nvar++;
  }

Par_rd = lookup_premia_enum_par(&(pt->flat_flag_rd), 1);
if (Par_rd[0].Val.V_FILENAME == NULL)
  {
    if ((Par_rd[0].Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH_LEN)) == NU
      return MEMORY_ALLOCATION_FAILURE;
    sprintf(Par_rd[0].Val.V_FILENAME, "%s%sinitialyield.dat", premia_data_dir,
  }

Par_rf = lookup_premia_enum_par(&(pt->flat_flag_rf), 1);
if (Par_rf[0].Val.V_FILENAME == NULL)
  {
    if ((Par_rf[0].Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH_LEN)) == NU
      return MEMORY_ALLOCATION_FAILURE;
    sprintf(Par_rf[0].Val.V_FILENAME, "%s%sinitialyield.dat", premia_data_dir,
  }

if (pt->sigma.Val.V_PNLVECT == NULL)
  {
    if ((pt->sigma.Val.V_PNLVECT = pnl_vect_create_from_double(3, 0.03)) == NU
      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->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 heshw2d;
MAKEMOD(heshw2d);
```