

[Help](#)

```
#include "
href../../mod/mrc30d/mrc30d_h_src.pdfmrc30d.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 "premia_obj.h"
static int MOD(Init)(Model *model)
{
    TYPEMOD *pt = (TYPEMOD *)(model->TypeModel);

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

        pt->Size.Vname = "Size";
        pt->Size.Vtype = PINT;
        pt->Size.Val.V_PINT = 30;
        pt->Size.Viter = FORBID;
        pt->Size.Vsetable = UNSETABLE;
        model->nvar++;

        pt->T.Vname = "Current Date";
        pt->T.Vtype = DATE;
        pt->T.Val.V_DATE = 0.;
        pt->T.Viter = ALLOW;
        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->kappa.Vname = "kappa";
```

```

pt->kappa.Vtype = DOUBLE;
pt->kappa.Val.V_DOUBLE = 1;
pt->kappa.Viter = ALLOW;
model->nvar++;

pt->eta.Vname = "eta";
pt->eta.Vtype = DOUBLE;
pt->eta.Val.V_DOUBLE = 1;
pt->eta.Viter = ALLOW;
model->nvar++;

pt->gama.Vname = "gamma";
pt->gama.Vtype = DOUBLE;
pt->gama.Val.V_DOUBLE = 8;
pt->gama.Viter = ALLOW;
model->nvar++;

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

pt->InitialStocksWeights.Vname = "InitialStocksWeights";
pt->InitialStocksWeights.Vtype = FILENAME;
pt->InitialStocksWeights.Val.V_FILENAME = NULL;
pt->InitialStocksWeights.Viter = FORBID;
pt->InitialStocksWeights.Vsetable = SETABLE;

model->nvar++;
if ((pt->InitialStocksWeights.Val.V_FILENAME = malloc(sizeof(char) * MAX_P
    return MEMORY_ALLOCATION_FAILURE;
sprintf(pt->InitialStocksWeights.Val.V_FILENAME, "%s%sInitialStocksWeights

pt->LocalVolatilities.Vname = "LocalVolatilities";
pt->LocalVolatilities.Vtype = FILENAME;
pt->LocalVolatilities.Val.V_FILENAME = NULL;
pt->LocalVolatilities.Viter = FORBID;
pt->LocalVolatilities.Vsetable = SETABLE;

model->nvar++;

```

```

        if ((pt->LocalVolatilities.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH))
            return MEMORY_ALLOCATION_FAILURE;
        sprintf(pt->LocalVolatilities.Val.V_FILENAME, "%s%sLocalVolatilities.dat",

pt->Basket_Correlation.Vname = "Basket_Correlation";
pt->Basket_Correlation.Vtype = FILENAME;
pt->Basket_Correlation.Val.V_FILENAME = NULL;
pt->Basket_Correlation.Viter = FORBID;
pt->Basket_Correlation.Vsetable = SETABLE;

model->nvar++;
if ((pt->Basket_Correlation.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH))
    return MEMORY_ALLOCATION_FAILURE;
sprintf(pt->Basket_Correlation.Val.V_FILENAME, "%s%sBasket_Correlation.dat",

pt->BasketLocalVolatility.Vname = "BasketLocalVolatility";
pt->BasketLocalVolatility.Vtype = FILENAME;
pt->BasketLocalVolatility.Val.V_FILENAME = NULL;
pt->BasketLocalVolatility.Viter = FORBID;
pt->BasketLocalVolatility.Vsetable = SETABLE;

model->nvar++;
if ((pt->BasketLocalVolatility.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH))
    return MEMORY_ALLOCATION_FAILURE;
sprintf(pt->BasketLocalVolatility.Val.V_FILENAME, "%s%sBasketLocalVolatility.dat",

    }

    return OK;
}

TYPEMOD mrc30d;
MAKEMOD(mrc30d);

```