

[Help](#)

```
#include "
href../../mod/heshw1d/heshw1d_h_src.pdfheshw1d.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"
#include "
href../../common/enums_h_src.pdfenums.h"

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

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

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

    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 = "Compound Dividend Rate";
pt->divid.Vtype = PDOUBLE;
pt->divid.Val.V_PDOUBLE = 0.03;
pt->divid.Viter = ALLOW;
model->nvar++;

pt->flat_flag.Vname = "Initial Yield Curve";
pt->flat_flag.Vtype = ENUM;
pt->flat_flag.Val.V_ENUM.value = 0;
pt->flat_flag.Val.V_ENUM.members = &PremiaEnumFlat;
pt->flat_flag.Viter = ALLOW;
model->nvar++;
Par = lookup_premia_enum_par(&(pt->flat_flag), 0);
Par[0].Vname = "Yield Value";
Par[0].Vtype = PDOUBLE;
Par[0].Val.V_PDOUBLE = 0.04;
Par[0].Viter = ALLOW;
Par = lookup_premia_enum_par(&(pt->flat_flag), 1);
Par[0].Vname = "Yield Curve";
Par[0].Vtype = FILENAME;
Par[0].Val.V_FILENAME = NULL;
Par[0].Viter = FORBID;

pt->kr.Vname = "Interest Rate Speed of Mean Reversion";
pt->kr.Vtype = PDOUBLE;
pt->kr.Val.V_PDOUBLE = 1;
pt->kr.Viter = ALLOW;
model->nvar++;

pt->Sigmar.Vname = "Interest Rate Volatility";
pt->Sigmar.Vtype = PDOUBLE;

```

```

pt->Sigmar.Val.V_PDOUBLE = 0.2;
pt->Sigmar.Viter = ALLOW;
model->nvar++;

pt->V0.Vname = "Current Variance";
pt->V0.Vtype = DOUBLE;
pt->V0.Val.V_DOUBLE = 0.1;
pt->V0.Viter = ALLOW;
model->nvar++;

pt->kV.Vname = "Speed of Mean Reversion Variance";
pt->kV.Vtype = DOUBLE;
pt->kV.Val.V_DOUBLE = 2;
pt->kV.Viter = ALLOW;
model->nvar++;

pt->thetaV.Vname = "Long-Run Variance";
pt->thetaV.Vtype = DOUBLE;
pt->thetaV.Val.V_DOUBLE = 0.1;
pt->thetaV.Viter = ALLOW;
model->nvar++;

pt->SigmaV.Vname = "Volatility of Variance";
pt->SigmaV.Vtype = DOUBLE;
pt->SigmaV.Val.V_DOUBLE = 0.3;
pt->SigmaV.Viter = ALLOW;
model->nvar++;

pt->RhoSr.Vname = "Rho S r";
pt->RhoSr.Vtype = RGDOUBLEM11;
pt->RhoSr.Val.V_RGDOUBLEM11 = -0.5;
pt->RhoSr.Viter = ALLOW;
model->nvar++;

pt->RhoSV.Vname = "Rho S V";
pt->RhoSV.Vtype = RGDOUBLEM11;
pt->RhoSV.Val.V_RGDOUBLEM11 = -0.5;
pt->RhoSV.Viter = ALLOW;
model->nvar++;

pt->RhorV.Vname = "Rho r V";

```

```

        pt->RhorV.Vtype = RGDOUBLEM11;
        pt->RhorV.Val.V_RGDOUBLEM11 = 0.0;
        pt->RhorV.Viter = ALLOW;
        model->nvar++;
    }

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

    return OK;
}

TYPEMOD heshw1d;
MAKEMOD(heshw1d);

```