

[Help](#)

```
#include "
href../..mod/hullwhite1dgeneralized/hullwhite1dgeneralized_h_src.pdfhullwhite1d
#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"

static PremiaEnumMember capletcurve_members[] =
{
    {"Caplet Volatility Curve 1", 1},
    {"Caplet Volatility Curve 2", 2},
    {NULL, NULLINT}
};

static DEFINE_ENUM(capletcurve, capletcurve_members);

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

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 = 0.0;
    pt->T.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 = "Initial Yield";
    Par[0].Vtype = PDOUBLE;
    Par[0].Val.V_PDOUBLE = 0.05;
    Par[0].Viter = FORBID;
    Par[0].Vsetable = SETABLE;
    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->CapletCurve.Vname = "Caplet Curve";
    pt->CapletCurve.Vtype = ENUM;
    pt->CapletCurve.Val.V_ENUM.value = 1;
    pt->CapletCurve.Val.V_ENUM.members = &capletcurve;
    pt->CapletCurve.Viter = ALLOW;
    model->nvar++;

    pt->a.Vname = "Speed of Mean Reversion";
    pt->a.Vtype = DOUBLE;
    pt->a.Val.V_DOUBLE = 0.1;

```

```

    pt->a.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 HullWhite1dGeneralized;
MAKEMOD(HullWhite1dGeneralized);

```