

[Help](#)

```
#include "hullwhite2d.h"
#include "chk.h"
#include "model.h"
#include "error_msg.h"
#include " premia_obj.h"
#include "enums.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 = 0.0;
        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 r";
Par[0].Vtype = PDOUBLE;
Par[0].Val.V_PDOUBLE = 0.03;
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->InitialYieldsu.Vname = "Initial u";
pt->InitialYieldsu.Vtype = PDOUBLE;
pt->InitialYieldsu.Val.V_PDOUBLE = 0.0;
pt->InitialYieldsu.Viter = ALLOW;
model->nvar++;

pt->aR.Vname = "Mean Reversion of r";
pt->aR.Vtype = PDOUBLE;
pt->aR.Val.V_PDOUBLE = 1.;
pt->aR.Viter = ALLOW;
model->nvar++;

pt->SigmaR.Vname = "Volatility of r";
pt->SigmaR.Vtype = PDOUBLE;
pt->SigmaR.Val.V_PDOUBLE = 0.01;
pt->SigmaR.Viter = ALLOW;
model->nvar++;

pt->bu.Vname = "Mean Reversion of u";
pt->bu.Vtype = PDOUBLE;
pt->bu.Val.V_PDOUBLE = 0.1;
pt->bu.Viter = ALLOW;
model->nvar++;
```

```

    pt->Sigmau.Vname = "Volatility of u";
    pt->Sigmau.Vtype = PDOUBLE;
    pt->Sigmau.Val.V_PDOUBLE = 0.0145;
    pt->Sigmau.Viter = ALLOW;
    model->nvar++;

    pt->Rho.Vname = "Rho";
    pt->Rho.Vtype = DOUBLE;
    pt->Rho.Val.V_DOUBLE = 0.6;
    pt->Rho.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 HullWhite2d;
MAKEMOD(HullWhite2d);

```