

[Help](#)

```
#include "
href../../mod/sg1d/sg1d_h_src.pdfsg1d.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 = 0.0;
    }
}
```

```

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 = "Yield Value";
Par[0].Vtype = PDOUBLE;
Par[0].Val.V_PDOUBLE = 0.05;
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->a.Vname = "Speed of Mean Reversion";
pt->a.Vtype = DOUBLE;
pt->a.Val.V_DOUBLE = 0.1;
pt->a.Viter = ALLOW;
model->nvar++;

pt->Sigma.Vname = "Short Rate Volatility";
pt->Sigma.Vtype = PDOUBLE;
pt->Sigma.Val.V_PDOUBLE = 0.05;
pt->Sigma.Viter = ALLOW;
model->nvar++;

model->HelpFilenameHint = "SG1D";
}

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\\sinitialeyield.dat", premia_data_dir, pa

```

```
    }  
  
    return OK;  
}  
TYPEMOD SquaredGaussian1d;  
MAKEMOD(SquaredGaussian1d);
```