```
   Help
#include "exoi.h"
#include "error_msg.h"

extern char* path_sep;


int OPT(Get)(int user,Planning *pt_plan,Option *opt, Model
    *mod)
{
    TYPEOPT* pt=(  TYPEOPT*)(opt->TypeOpt);

    (opt->Init)(opt, mod);

    if (user==TOSCREEN)
        if  ((opt->Show)(user,pt_plan,opt,mod))
            do
            {
                Fprintf(TOSCREEN,"_________________________
    ____Option:%s{n",opt->Name);

                if ((strcmp(opt->Name,"    CallableCappedFloater")==0))
                {
                    ScanVar(pt_plan,user,&(pt->FirstExercis
    eDate));
                    ScanVar(pt_plan,user,&(pt->LastPaymentD
    ate));
                    ScanVar(pt_plan,user,&(pt->ResetPeriod)
    );
                    ScanVar(pt_plan,user,&(pt->Nominal));
                    ScanVar(pt_plan,user,&(pt->Spread));
                    ScanVar(pt_plan,user,&(pt->Cap));
                }
                if ((strcmp(opt->Name,"    CallableInverseFloater")==0))
                {
                    ScanVar(pt_plan,user,&(pt->FirstExercis
    eDate));
                    ScanVar(pt_plan,user,&(pt->LastPaymentD
    ate));
                    ScanVar(pt_plan,user,&(pt->ResetPeriod)
    );
```

```
                ScanVar(pt_plan,user,&(pt->Nominal));
                ScanVar(pt_plan,user,&(pt->Cap));
                ScanVar(pt_plan,user,&(pt->Strike));
                ScanVar(pt_plan,user,&(pt->Gearing));
                ScanVar(pt_plan,user,&(pt->Floor));
            }

            if ((strcmp(opt->Name,"    CallableRangeAccrual")==0))
            {
                ScanVar(pt_plan,user,&(pt->FirstExercis
eDate));
                ScanVar(pt_plan,user,&(pt->LastPaymentD
ate));
                ScanVar(pt_plan,user,&(pt->ResetPeriod)
);
                ScanVar(pt_plan,user,&(pt->Nominal));
                ScanVar(pt_plan,user,&(pt->FixedRate));
                ScanVar(pt_plan,user,&(pt->LowerRangeB
ound));
                ScanVar(pt_plan,user,&(pt->UpperRangeB
ound));
            }

            if ((strcmp(opt->Name,"CallableCMSSpread")=
=0))
            {
                ScanVar(pt_plan,user,&(pt->FirstExercis
eDate));
                ScanVar(pt_plan,user,&(pt->LastPaymentD
ate));
                ScanVar(pt_plan,user,&(pt->ResetPeriod)
);
                ScanVar(pt_plan,user,&(pt->Nominal));
                ScanVar(pt_plan,user,&(pt->Cap));
                ScanVar(pt_plan,user,&(pt->Floor));
                ScanVar(pt_plan,user,&(pt->CMSMat1));
                ScanVar(pt_plan,user,&(pt->CMSMat2));
            }
        }
        while ((opt->Show)(user,pt_plan,opt,mod));
```

```
    return (opt->Show)(TOSCREENANDFILE,pt_plan,opt,mod);
}


int OPT(FGet)(char **InputFile,int user,Planning *pt_plan,
    Option *opt, Model *mod)
{
    TYPEOPT* pt=(  TYPEOPT*)(opt->TypeOpt);

    (opt->Init)(opt, mod);

    if (user==TOSCREEN)
    {
        Fprintf(TOSCREEN,"_____________________________
    Option:%s{n",opt->Name);

        if (strcmp(opt->Name,"CallableCappedFloater")==0)
        {
            FScanVar(InputFile,pt_plan,user,&(pt->FirstEx
    erciseDate));
            FScanVar(InputFile,pt_plan,user,&(pt->LastPay
    mentDate));
            FScanVar(InputFile,pt_plan,user,&(pt->Nominal))
    ;
            FScanVar(InputFile,pt_plan,user,&(pt->ResetPe
    riod));
            FScanVar(InputFile,pt_plan,user,&(pt->Spread));
            FScanVar(InputFile,pt_plan,user,&(pt->Cap));
        }
        if (strcmp(opt->Name,"CallableInverseFloater")==0)
        {
            FScanVar(InputFile,pt_plan,user,&(pt->FirstEx
    erciseDate));
            FScanVar(InputFile,pt_plan,user,&(pt->LastPay
    mentDate));
            FScanVar(InputFile,pt_plan,user,&(pt->ResetPe
    riod));
            FScanVar(InputFile,pt_plan,user,&(pt->Nominal))
    ;
            FScanVar(InputFile,pt_plan,user,&(pt->Cap));
            FScanVar(InputFile,pt_plan,user,&(pt->Strike));
        FScanVar(InputFile,pt_plan,user,&(pt->Gearing))
;
        FScanVar(InputFile,pt_plan,user,&(pt->Floor));
    }

    if (strcmp(opt->Name,"CallableRangeAccrual")==0)
    {
        FScanVar(InputFile,pt_plan,user,&(pt->FirstEx
erciseDate));
        FScanVar(InputFile,pt_plan,user,&(pt->LastPay
mentDate));
        FScanVar(InputFile,pt_plan,user,&(pt->ResetPe
riod));
        FScanVar(InputFile,pt_plan,user,&(pt->Nominal))
;
        FScanVar(InputFile,pt_plan,user,&(pt->FixedRa
te));
        FScanVar(InputFile,pt_plan,user,&(pt->LowerRang
eBound));
        FScanVar(InputFile,pt_plan,user,&(pt->UpperRan
geBound));
    }

    if (strcmp(opt->Name,"CallableCMSSpread")==0)
    {
        FScanVar(InputFile,pt_plan,user,&(pt->FirstEx
erciseDate));
        FScanVar(InputFile,pt_plan,user,&(pt->LastPay
mentDate));
        FScanVar(InputFile,pt_plan,user,&(pt->ResetPe
riod));
        FScanVar(InputFile,pt_plan,user,&(pt->Nominal))
;
        FScanVar(InputFile,pt_plan,user,&(pt->Cap));
        FScanVar(InputFile,pt_plan,user,&(pt->Floor));
        FScanVar(InputFile,pt_plan,user,&(pt->CMSMat1))
;
        FScanVar(InputFile,pt_plan,user,&(pt->CMSMat2))
;
    }
    }

    return (opt->Show)(TOSCREENANDFILE,pt_plan,opt,mod);
}


int OPT(Show)(int user,Planning *pt_plan,Option *opt,
    Model *mod)
{
    TYPEOPT* pt=(TYPEOPT*)(opt->TypeOpt);

    (void)(opt->Init)(opt, mod);
    Fprintf(user,"##Option:%s{n",opt->Name);
    /* Valid Parameters*/
    if ((strcmp(opt->Name,"CallableCappedFloater")==0))
    {
        PrintVar(pt_plan,user,&(pt->FirstExerciseDate));
        PrintVar(pt_plan,user,&(pt->LastPaymentDate));
        PrintVar(pt_plan,user,&(pt->ResetPeriod));
        PrintVar(pt_plan,user,&(pt->Nominal));
        PrintVar(pt_plan,user,&(pt->Spread));
        PrintVar(pt_plan,user,&(pt->Cap));

    }

    if ((strcmp(opt->Name,"CallableInverseFloater")==0))
    {
        PrintVar(pt_plan,user,&(pt->FirstExerciseDate));
        PrintVar(pt_plan,user,&(pt->LastPaymentDate));
        PrintVar(pt_plan,user,&(pt->ResetPeriod));
        PrintVar(pt_plan,user,&(pt->Nominal));
        PrintVar(pt_plan,user,&(pt->Cap));
        PrintVar(pt_plan,user,&(pt->Strike));
        PrintVar(pt_plan,user,&(pt->Gearing));
        PrintVar(pt_plan,user,&(pt->Floor));
    }

    if ((strcmp(opt->Name,"CallableRangeAccrual")==0))
    {
        PrintVar(pt_plan,user,&(pt->FirstExerciseDate));
        PrintVar(pt_plan,user,&(pt->LastPaymentDate));
        PrintVar(pt_plan,user,&(pt->ResetPeriod));
        PrintVar(pt_plan,user,&(pt->Nominal));
        PrintVar(pt_plan,user,&(pt->FixedRate));
        PrintVar(pt_plan,user,&(pt->LowerRangeBound));
        PrintVar(pt_plan,user,&(pt->UpperRangeBound));
    }

    if ((strcmp(opt->Name,"CallableCMSSpread")==0))
    {
        PrintVar(pt_plan,user,&(pt->FirstExerciseDate));
        PrintVar(pt_plan,user,&(pt->LastPaymentDate));
        PrintVar(pt_plan,user,&(pt->ResetPeriod));
        PrintVar(pt_plan,user,&(pt->Nominal));
        PrintVar(pt_plan,user,&(pt->Cap));
        PrintVar(pt_plan,user,&(pt->Floor));
        PrintVar(pt_plan,user,&(pt->CMSMat1));
        PrintVar(pt_plan,user,&(pt->CMSMat2));
    }

    return (opt->Check)(user,pt_plan,opt);
}



extern Option OPT(CallableCappedFloater);
extern Option OPT(CallableInverseFloater);
extern Option OPT(CallableRangeAccrual);
extern Option OPT(CallableCMSSpread);

Option* OPT(family)[]=
{
    &OPT(CallableCappedFloater),
    &OPT(CallableInverseFloater),
    &OPT(CallableRangeAccrual),
    &OPT(CallableCMSSpread),
    NULL
};
```

References