

```

    Help
#include "lim.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);

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

    if (user==TOSCREEN)
        if ((opt->Show)(user,pt_plan,opt,mod))
            do
            {
                Fprintf(TOSCREEN,"-----
Option:%s\n",opt->Name);
                ScanVar(pt_plan,user,&(pt->Maturity));
                GetParVar(pt_plan,user,(pt->Limit.Val.V_NUMFUNC_1
)->Par);
                GetParVar(pt_plan,user,(pt->PayOff.Val.V_NUMFUNC_
1)->Par);
                if ((pt->RebOrNo).Val.V_BOOL==REBATE)
                    GetParVar(pt_plan,user,(pt->Rebate.Val.V_
NUMFUNC_1)->Par);
            }
            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);

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

    if (user==TOSCREEN)

```

```

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

        FScanVar(InputFile, pt_plan, user, &(pt->Maturity));
        FGetParVar(InputFile, pt_plan, user, (pt->Limit.Val.V_
NUMFUNC_1)->Par);
        FGetParVar(InputFile, pt_plan, user, (pt->PayOff.Val.V_
NUMFUNC_1)->Par);
        if ((pt->RebOrNo).Val.V_BOOL==REBATE)
FGetParVar(InputFile, pt_plan, user, (pt->Rebate.Val.V_
NUMFUNC_1)->Par);
    }

    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);

    PrintVar(pt_plan, user, &(pt->Maturity));
    ShowParVar(pt_plan, user, (pt->Limit.Val.V_NUMFUNC_1)->Par)
    ;
    ShowParVar(pt_plan, user, (pt->PayOff.Val.V_NUMFUNC_1)->
Par);
    if ((pt->RebOrNo).Val.V_BOOL==REBATE)
        ShowParVar(pt_plan, user, (pt->Rebate.Val.V_NUMFUNC_1)->
Par);

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

extern Option OPT(PutUpOutEuro);
extern Option OPT(CallDownOutAmer);

```

```

extern Option OPT(CallDownOutEuro);
extern Option OPT(CallUpInEuro);
extern Option OPT(CallUpInAmer);
extern Option OPT(CallUpOutAmer);
extern Option OPT(CallUpOutEuro);
extern Option OPT(PutDownInEuro);
extern Option OPT(PutDownInAmer);
extern Option OPT(PutDownOutAmer);
extern Option OPT(PutDownOutEuro);
extern Option OPT(PutUpInEuro);
extern Option OPT(PutUpInAmer);
extern Option OPT(PutUpOutAmer);
extern Option OPT(CallDownInEuro);
extern Option OPT(CallDownInAmer);
extern Option OPT(ParisianCallDownOutEuro);
extern Option OPT(ParisianCallDownInEuro);
extern Option OPT(ParisianPutDownOutEuro);
extern Option OPT(ParisianPutDownInEuro);
extern Option OPT(ParisianCallUpOutEuro);
extern Option OPT(ParisianCallUpInEuro);
extern Option OPT(ParisianPutUpOutEuro);
extern Option OPT(ParisianPutUpInEuro);

```

```

Option* OPT(family)[]={
    &OPT(CallDownOutEuro),
    &OPT(PutDownOutEuro),
    &OPT(CallUpOutEuro),
    &OPT(PutUpOutEuro),
    &OPT(CallDownInEuro),
    &OPT(CallUpInEuro),
    &OPT(PutDownInEuro),
    &OPT(PutUpInEuro),
    &OPT(CallDownOutAmer),
    &OPT(PutDownOutAmer),
    &OPT(CallUpOutAmer),
    &OPT(PutUpOutAmer),
    &OPT(CallDownInAmer),
    &OPT(PutDownInAmer),
    &OPT(CallUpInAmer),
    &OPT(PutUpInAmer),
    &OPT(ParisianCallDownOutEuro),

```

```
&OPT(ParisianCallDownInEuro),  
&OPT(ParisianPutDownOutEuro),  
&OPT(ParisianPutDownInEuro),  
&OPT(ParisianCallUpOutEuro),  
&OPT(ParisianCallUpInEuro),  
&OPT(ParisianPutUpOutEuro),  
&OPT(ParisianPutUpInEuro),  
NULL  
};
```

References