

[Help](#)

```

#include "lmm1d_exoi.h"

int MOD_OPT(ChkMix)(Option *Opt, Model *Mod)
{
    TYPEOPT *ptOpt = (TYPEOPT *) (Opt->TypeOpt);
    TYPEMOD *ptMod = (TYPEMOD *) (Mod->TypeModel);
    int status = OK;

    if ((strcmp(Opt->Name, "CallableCappedFloater") == 0) || (strcmp(Opt->Name, "C
    {
        if ((ptOpt->FirstExerciseDate.Val.V_DATE) <= (ptMod->T.Val.V_DATE))
        {
            Fprintf(TOSCREENANDFILE, "Current date greater than first exercise dat
            status += 1;
        }
        if ((ptOpt->FirstExerciseDate.Val.V_DATE) >= (ptOpt->LastPaymentDate.Val.V
        {
            Fprintf(TOSCREENANDFILE, "First exercise date greater than last paymen
            status += 1;
        }
    }

    return status;
}

extern PricingMethod MET(MC_LongstaffSchwartz_CallableCappedFloater);
extern PricingMethod MET(MC_LongstaffSchwartz_CallableInverseFloater);
extern PricingMethod MET(MC_LongstaffSchwartz_CallableRangeAccrual);
extern PricingMethod MET(MC_LongstaffSchwartz_CallableCMSSpread);

PricingMethod *MOD_OPT(methods)[] =
{
    &MET(MC_LongstaffSchwartz_CallableCappedFloater),
    &MET(MC_LongstaffSchwartz_CallableInverseFloater),
    &MET(MC_LongstaffSchwartz_CallableRangeAccrual),
    &MET(MC_LongstaffSchwartz_CallableCMSSpread),

```

```
    NULL
};
DynamicTest *MOD_OPT(tests) [] =
{
    NULL
};
```

```
Pricing MOD_OPT(pricing) =
{
    ID_MOD_OPT,
    MOD_OPT(methods),
    MOD_OPT(tests),
    MOD_OPT(ChkMix)
};
```