

[Help](#)

```
#include "
href../../../../mod/lmm_stochvol_piterbarg/lmm_stochvol_piterbarg_std/lmm_stochvol

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, "PayerSwaption") == 0) || (strcmp(Opt->Name, "ReceiverS
        if ((ptOpt->BMaturity.Val.V_DATE) <= (ptOpt->OMaturity.Val.V_DATE))
        {
            Fprintf(TOSCREENANDFILE, "Option maturity greater than Bond maturity!\n n
            status += 1;
        }

    if ((strcmp(Opt->Name, "Floor") == 0) || (strcmp(Opt->Name, "Cap") == 0))
    {

        if ((ptOpt->FirstResetDate.Val.V_DATE) <= (ptMod->T.Val.V_DATE))
        {
            Fprintf(TOSCREENANDFILE, "Current date greater than first coupon date!
            status += 1;
        }
        if ((ptOpt->FirstResetDate.Val.V_DATE) >= (ptOpt->BMaturity.Val.V_DATE))
        {
            Fprintf(TOSCREENANDFILE, "First reset date greater than contract matur
            status += 1;
        }
    }

    return status;
}

extern PricingMethod MET(AP_Swaption_LmmPit);
extern PricingMethod MET(AP_CaplFloor_LmmPit);

PricingMethod *MOD_OPT(methods)[] =
```

```

{

    &MET(AP_Swaption_LmmPit),
    &MET(AP_CaplFloor_LmmPit),

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

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

```