

[Help](#)

```
#include "
href../../../../mod/temperedstable1d/temperedstable1d_vol/temperedstable1d_vol_h_sr

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

    if ((ptOpt->Maturity.Val.V_DATE) <= (ptMod->T.Val.V_DATE))
    {
        Fprintf(TOSCREENANDFILE, "Current date greater than maturity!\ n");
        status += 1;
    };

    return status;
}

extern PricingMethod MET(AP_KELLERRESSEL);
extern PricingMethod MET(AP_TSL_FFT);
extern PricingMethod MET(AP_CGMY_REALVAR);
extern PricingMethod MET(CF_CGMY_VARIANCESWAP);
extern PricingMethod MET(AP_REPL1_VARIANCESWAP);
extern PricingMethod MET(AP_REPL2_VARIANCESWAP);
extern PricingMethod MET(AP_CGMY_VOLATILITYSWAP);

PricingMethod *MOD_OPT(methods) [] =
{
    &MET(AP_KELLERRESSEL),
    &MET(AP_TSL_FFT),
    &MET(AP_CGMY_REALVAR),
    &MET(CF_CGMY_VARIANCESWAP),
    &MET(AP_REPL1_VARIANCESWAP),
    &MET(AP_REPL2_VARIANCESWAP),
    &MET(AP_CGMY_VOLATILITYSWAP),
    NULL
};

DynamicTest *MOD_OPT(tests) [] =
```

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