

## [Help](#)

```
#include "
href../../mod/nonpar1d/nonpar1d_vol/nonpar1d_vol_h_src.pdfnonpar1d_vol.h"

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_NONPAR_VARIANCESWAP);
extern PricingMethod MET(AP_NONPAR_VOLATILITYSWAP);
extern PricingMethod MET(AP_NONPAR_VOLATILITYSWAP1);
extern PricingMethod MET(AP_NONPAR_VOLATILITYINDEX);
extern PricingMethod MET(AP_NONPAR_LEVY_VOLATILITYINDEX);
extern PricingMethod MET(AP_NONPAR_LEVY_VOLATILITYSWAP);

//extern PricingMethod MET(AP_NONPAR_REALVAROPTIONS);

PricingMethod *MOD_OPT(methods) [] =
{
    &MET(AP_NONPAR_VARIANCESWAP),
    &MET(AP_NONPAR_VOLATILITYSWAP),
    &MET(AP_NONPAR_VOLATILITYSWAP1),
    &MET(AP_NONPAR_VOLATILITYINDEX),
    &MET(AP_NONPAR_LEVY_VOLATILITYINDEX),
    &MET(AP_NONPAR_LEVY_VOLATILITYSWAP),
    //&MET(AP_NONPAR_REALVAROPTIONS),
    NULL
};
```

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

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