

[Help](#)

```
#include "
href../../../../mod/kould/kould_std/kould_std_h_src.pdfkould_std.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_fastwhamerdig_kou);
extern PricingMethod MET(AP_fastwhamer_kou);
extern PricingMethod MET(AP_CarrKou);
extern PricingMethod MET(AP_Kou_Eu);
extern PricingMethod MET(AP_Kou_Am);
extern PricingMethod MET(AP_KIRKBY_KOU);
extern PricingMethod MET(MC_Kou);
extern PricingMethod MET(FD_ImpExp);
extern PricingMethod MET(MC_Kou_Digital_LRM);

PricingMethod *MOD_OPT(methods)[] =
{
    &MET(AP_fastwhamer_kou),
    &MET(AP_fastwhamerdig_kou),
    &MET(FD_ImpExp),
    &MET(AP_CarrKou),
    &MET(AP_Kou_Eu),
    &MET(AP_Kou_Am),
    &MET(AP_KIRKBY_KOU),
    &MET(MC_Kou),
}
```

```

    &MET(MC_Kou_Digital_LRM),
    NULL
};

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

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

```