

Help

```

#include "bsnd_stdnd.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_CarmonaDurrleman);
extern PricingMethod MET(MC_BSDE_Labart);
extern PricingMethod MET(MC_Jourdain_Lelong);
extern PricingMethod MET(MC_LongstaffSchwartzND);
extern PricingMethod MET(MC_TsitsiklisVanRoyND);
extern PricingMethod MET(MC_BarraquandMartineauND);
extern PricingMethod MET(MC_QuantizationND);
extern PricingMethod MET(MC_RandomQuantizationND);
extern PricingMethod MET(MC_QuantizationStoredND);
extern PricingMethod MET(MC_BroadieGlassermannND);
//extern PricingMethod MET(FD_LeentvaarOosterlee);
extern PricingMethod MET(MC_AndersenBroadieND);
extern PricingMethod MET(MC_MalliavinAmer);

PricingMethod* MOD_OPT(methods)[]={
    &MET(AP_CarmonaDurrleman),
    &MET(MC_BSDE_Labart),
    &MET(MC_Jourdain_Lelong),
    &MET(MC_LongstaffSchwartzND),
    &MET(MC_TsitsiklisVanRoyND),
    &MET(MC_BarraquandMartineauND),
    &MET(MC_QuantizationND),
    &MET(MC_RandomQuantizationND),

```

```
&MET(MC_QuantizationStoredND),  
&MET(MC_BroadieGlassermannND),  
&MET(MC_AndersenBroadieND),  
&MET(MC_MalliavinAmer),  
//&MET(FD_LeentvaarOosterlee),  
NULL  
};
```

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

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

References