

[Help](#)

```
#include "local_vol_callable.h"
#include "enums.h"
#include "error_msg.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_basis.h"

#if defined(PremiaCurrentVersion) && PremiaCurrentVersion < (2011+2) //The "#els

int CALC(MC_RIBSDE)(void *Opt, void *Mod, PricingMethod *Met)
{
    return AVAILABLE_IN_FULL_PREMIA;
}

static int CHK_OPT(MC_RIBSDE)(void *Opt, void *Mod)
{
    return NONACTIVE;
}

static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    if (Met->init == 0)
    {
        Met->init = 1;
    }

    return OK;
}

#else

static int(*price_func)(double *prix, TYPEMOD *Mod, TYPEOPT *Opt, int gen, int b

extern int callable_highly_path_dep_protection(double *prix, TYPEMOD *Mod, TYPEO
extern int callable_intermittent_protection(double *prix, TYPEMOD *Mod, TYPEOPT
extern int callable_no_call_protection(double *prix, TYPEMOD *Mod, TYPEOPT *Opt,
extern int callable_path_dep_protection(double *prix, TYPEMOD *Mod, TYPEOPT *Op
extern int callable_std_protection(double *prix, TYPEMOD *Mod, TYPEOPT *Opt, int

int CALC(MC_RIBSDE)(void *Opt, void *Mod, PricingMethod *Met)
```

```

{
    TYPEOPT *ptOpt = (TYPEOPT *)Opt;
    TYPEMOD *ptMod = (TYPEMOD *)Mod;

    if (price_func == NULL)
    {
        printf("Some initialization is missing");
        return FAIL;
    }
    (*price_func)(&(Met->Res[0].Val.V_DOUBLE), ptMod, ptOpt, Met->Par[1].Val.V_ENU
    return OK;
}

static int CHK_OPT(MC_RIBSDE)(void *Opt, void *Mod)
{
    return OK;
}

static int MET(Init)(PricingMethod *Met, Option *Opt)
{
    if (Met->init == 0)
    {
        Met->Par[0].Val.V_LONG = 10000;
        Met->Par[1].Val.V_ENUM.value = 0;
        Met->Par[1].Val.V_ENUM.members = &PremiaEnumMCRNGs;
        Met->Par[2].Val.V_ENUM.value = 0;
        Met->Par[2].Val.V_ENUM.members = &PremiaEnumBasis;
        Met->Par[3].Val.V_INT = 4;
        Met->Par[4].Val.V_INT = 4;

        Met->init = 1;
    }

    if (Opt)
    {
        if (strcmp(Opt->Name, "NoCall") == 0) price_func = callable_no_call_protec
        if (strcmp(Opt->Name, "StdCall") == 0) price_func = callable_std_protectio
        if (strcmp(Opt->Name, "PathDepCall") == 0) price_func = callable_path_dep_
        if (strcmp(Opt->Name, "HighlyPathDepCall") == 0) price_func = callable_hig
        if (strcmp(Opt->Name, "IntermittentCall") == 0) price_func = callable_inte
    }
}

```

```
    return OK;
}

#endif //PremiaCurrentVersion

PricingMethod MET(MC_RIBSDE) =
{
    "MC_RIBSDE",
    { {"Nb iterations", LONG, {1000}, ALLOW},
      {"RandomGenerator", ENUM, {0}, ALLOW},
      {"Basis", ENUM, {1}, ALLOW},
      {"Basis Dimension", INT, {100}, ALLOW},
      {"Nb discretization steps per day", INT, {4}, ALLOW},
      {" ", PREMIA_NULLTYPE, {0}, FORBID}
    },
    CALC(MC_RIBSDE),
    {{"Price", DOUBLE, {100}, FORBID}, {" ", PREMIA_NULLTYPE, {0}, FORBID}},
    CHK_OPT(MC_RIBSDE),
    CHK_ok,
    MET(Init)
} ;
```