```
   Help
#include "optype.h"
#include "enums.h"
#include "pnl/pnl_random.h"
#include "pnl/pnl_basis.h"

static PremiaEnumMember BooleanMembers[] =
  {
    { "No",  0 },
    { "Yes", 1 },
    { NULL, NULLINT }
  };

static PremiaEnumMember CirOrderMembers[] =
{
  { "Second Order for the CIR", 1 },
  { "Third Order for the CIR", 2 },
  { NULL, NULLINT }
};

static PremiaEnumMember afd_members[] =
{
  { "Terminal Measure", 0 },
  { "Spot Measure", 1 },
  { NULL, NULLINT }
};

static PremiaEnumMember averaging_members[] =
{
  { "Averaged Vol", 0 },
  { "Time-Dep Vol", 1 },
  { NULL, NULLINT }
};

static PremiaEnumMember boundary_cond_members[] =
{
  {"Dirichlet", 0},
  {"Andreasen", 1},
  { NULL, NULLINT }
};
```

```
static PremiaEnumMember DiscretizationScheme_members[] =
{
  { "Exact Scheme for Wishart and Weak Scheme for Stock", 1
     },
  { "Weak Scheme for Stock and Wishart", 2 },
  { NULL, NULLINT }
};

static PremiaEnumMember PrecondMembers[] =
{
  { "Diagonal", 1 },
  { "ILU", 2 },
  { NULL, NULLINT }
};

static PremiaEnumMember schemetreenig_members[] =
{
  { "Improved Scheme", 1 },
  { "MSS Scheme", 2 },
  { NULL, NULLINT }
};

static PremiaEnumMember exp_part_members[] =
{
  { "Decentered", 1 },
  { "Centered", 2 },
  { NULL, NULLINT }
};

static PremiaEnumMember DeltaMethodMembers[] =
{
  { "Finite Difference", 1 },
  { "Malliavin", 2 },
  { "Malliavin Local", 3 },
  { NULL, NULLINT }
};

static PremiaEnumMember IntegralSchemeMembers[] =
{
  { "Riemann", 1 },
  { "Trapezoidal", 2 },
```

```
  { "Brownian Bridge", 3 },
  { NULL, NULLINT }
};

static PremiaEnumMember PnlBasisMembers [] =
  {
    { "Canonical", PNL_BASIS_CANONICAL},
    { "Hermite", PNL_BASIS_HERMITIAN},
    { "Tchebychev", PNL_BASIS_TCHEBYCHEV},
    { NULL, NULLINT},
  };

/*
 * Random Number Generator Array
 */
static PremiaEnumMember PnlRngMembers[]=
  {
    {"KNUTH", PNL_RNG_KNUTH},
    {"MRGK3", PNL_RNG_MRGK3},
    {"MRGK5", PNL_RNG_MRGK5},
    {"SHUFL", PNL_RNG_SHUFL},
    {"L'ECUYER", PNL_RNG_LECUYER},
    {"TAUSWORTHE", PNL_RNG_TAUSWORTHE},
    {"MERSENNE", PNL_RNG_MERSENNE},
    {"MERSENNE (Random Seed)", PNL_RNG_MERSENNE_RANDOM_SEED
    },
    {"SQRT", PNL_RNG_SQRT},
    {"HALTON", PNL_RNG_HALTON},
    {"FAURE", PNL_RNG_FAURE},
    {"SOBOL_I4", PNL_RNG_SOBOL_I4},
    {"SOBOL_I8", PNL_RNG_SOBOL_I8},
    {"NIEDERREITER", PNL_RNG_NIEDERREITER},
    {NULL, NULLINT}
  };

/*
 * True MC generators do not take into account the paramete
    r dimension in the
 * Compute function.
 */
static PremiaEnumMember PnlRngMCMembers[]=
```

```
  {
    {"KNUTH", PNL_RNG_KNUTH},
    {"MRGK3", PNL_RNG_MRGK3},
    {"MRGK5", PNL_RNG_MRGK5},
    {"SHUFL", PNL_RNG_SHUFL},
    {"L'ECUYER", PNL_RNG_LECUYER},
    {"TAUSWORTHE", PNL_RNG_TAUSWORTHE},
    {"MERSENNE", PNL_RNG_MERSENNE},
    {"MERSENNE (Random Seed)", PNL_RNG_MERSENNE_RANDOM_SEED
    },
    {NULL, NULLINT}
  };

static PremiaEnumMember flat_members[] =
{
    {"Flat",0, 1},
    {"No_Flat ZCB Prices in data/initialyield.dat",1, 0},
    { NULL, NULLINT, 0}
};

static PremiaEnumMember flat_members2[] =
{
    {"Flat",0, 1},
    {"No_Flat ZCB Prices in data/initialyield.dat",1, 1},
    { NULL, NULLINT, 0}
};


DEFINE_ENUM(PremiaEnumBool, BooleanMembers);
DEFINE_ENUM(PremiaEnumCirOrder,CirOrderMembers);
DEFINE_ENUM(PremiaEnumAfd, afd_members);
DEFINE_ENUM(PremiaEnumAveraging, averaging_members);
DEFINE_ENUM(PremiaEnumBoundaryCond, boundary_cond_members);
DEFINE_ENUM(PremiaEnumDiscretizationScheme,Discretization
    Scheme_members);
DEFINE_ENUM(PremiaEnumPrecond, PrecondMembers);
DEFINE_ENUM(PremiaEnumSchemeTreeMSS, schemetreenig_members)
    ;
DEFINE_ENUM(PremiaEnumExpPart, exp_part_members);
DEFINE_ENUM(PremiaEnumDeltaMC, DeltaMethodMembers);
DEFINE_ENUM(PremiaEnumIntegralScheme, IntegralSchemeMembers
```

```
    )
DEFINE_ENUM(PremiaEnumRNGs, PnlRngMembers);
DEFINE_ENUM(PremiaEnumMCRNGs, PnlRngMCMembers);
DEFINE_ENUM(PremiaEnumBasis, PnlBasisMembers);
DEFINE_ENUM(PremiaEnumFlat, flat_members);
DEFINE_ENUM(PremiaEnumFlat2, flat_members2);
```

References