

[Help](#)

```

#ifndef _ENUMS_H
#define _ENUMS_H

#include "optype.h"

#define NULLINT -1      /*!< default value of empty choices
    */
#define MAX_PAR_ENUM 2 /*!< maximum number of parameters fo
    r Enumerations */

/* the array of PremiaEnumMember must end with {NULL, NUL
    LINT, 0, 0} */
typedef struct PremiaEnumMember_t PremiaEnumMember;
struct PremiaEnumMember_t
{
    const char *label; /*!< string describing the choice */
    int        key;    /*!< value associated to this choice */
    int        nvar;   /*!< length of array Par, must be small
        er than MAX_PAR_ENUM */
    VAR        Par[MAX_PAR_ENUM]; /*!< extra parameters ass
        ociated to this choice */
};

typedef struct PremiaEnum_t PremiaEnum;
struct PremiaEnum_t
{
    unsigned    size;      /*!< size in bytes of an
        enum member      */
    PremiaEnumMember *members; /*!< a pointer to the first
        member of the enum */
    const char    *label;   /*!< printable label for th
        e enumeration    */
};

#define DEFINE_ENUM(Name, Members) PremiaEnum Name = {
    { sizeof(Members[0]), &Members[0], #Name };

extern PremiaEnum PremiaEnumNull;

```

```
extern PremiaEnum PremiaEnumBool;
extern PremiaEnum PremiaEnumCirOrder;
extern PremiaEnum PremiaEnumAfd;
extern PremiaEnum PremiaEnumAveraging;
extern PremiaEnum PremiaEnumBoundaryCond;
extern PremiaEnum PremiaEnumDiscretizationScheme;
extern PremiaEnum PremiaEnumPrecond;
extern PremiaEnum PremiaEnumSchemeTreeMSS;
extern PremiaEnum PremiaEnumExpPart;
extern PremiaEnum PremiaEnumDeltaMC;
extern PremiaEnum PremiaEnumIntegralScheme;
extern PremiaEnum PremiaEnumFlat;
extern PremiaEnum PremiaEnumFlatCopy;
extern PremiaEnum PremiaEnumFlat2;
extern PremiaEnum PremiaEnumBasis;
extern PremiaEnum PremiaEnumRNGs;
extern PremiaEnum PremiaEnumMCRNGs;
extern PremiaEnum PremiaEnumVolatility;
extern PremiaEnum PremiaEnumISType;

/* defined in var.c, but convenient to pout it here */
extern VAR *lookup_premia_enum_par(const VAR *x, int key);

#endif /* _ENUMS_H */
```

References