

[Help](#)

```

#ifndef _ENUMS_H
#define _ENUMS_H

#include "optype.h"

#define NULLINT -1      /*!< default value of empty choices
    */
#define MAX_PAR_ENUM 1 /*!< 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 PremiaEnumBool;

```

```
extern PremiumEnum PremiumEnumCirOrder;
extern PremiumEnum PremiumEnumAfd;
extern PremiumEnum PremiumEnumAveraging;
extern PremiumEnum PremiumEnumBoundaryCond;
extern PremiumEnum PremiumEnumDiscretizationScheme;
extern PremiumEnum PremiumEnumPrecond;
extern PremiumEnum PremiumEnumSchemeTreeMSS;
extern PremiumEnum PremiumEnumExpPart;
extern PremiumEnum PremiumEnumDeltaMC;
extern PremiumEnum PremiumEnumIntegralScheme;
extern PremiumEnum PremiumEnumFlat;
extern PremiumEnum PremiumEnumFlat2;
extern PremiumEnum PremiumEnumBasis;
extern PremiumEnum PremiumEnumRNGs;
extern PremiumEnum PremiumEnumMCRNGs;

/* 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