

## [Help](#)

```
#ifndef _LIMDISC_H
#define _LIMDISC_H

#include "
href../../../../common/optype_h_src.pdfoptype.h"
#include "
href../../../../common/var_h_src.pdfvar.h"
#include "
href../../../../common/chk_h_src.pdfchk.h"
#include "
href../../../../common/numfunc_h_src.pdfnumfunc.h"
#include "
href../../../../common/option_h_src.pdfoption.h"

#define TYPEOPT LIMDISC

/*Limit Option// Single barrier*/

typedef struct TYPEOPT
{
    VAR Maturity;
    VAR Limit;      /*The Limit definition:
    * starting_date is in Limit->[0],
    * final_date(always equal to maturity for this family, so useless)is in Limit-
    * frequency is in Limit->Par[2],
    * the value of the limit is in Limit->Par[3]
    * !!!!!WARNING!!!!
    * Wether the limit is backard/forward
    * should be tested in ChkOpt
    */
    VAR PayOff;
    VAR Rebate;
    VAR OutOrIn;
    VAR DownOrUp;
    VAR RebOrNo;
    VAR EuOrAm;
    VAR PartOrTot; /*Partial Or Total limit
    * a partial limit is specified
    * by starting_date, final_date
```

```

*/
VAR ContOrDisc; /*Continuous or Discrete:
*   a discrete limit is specified
*   by frequency (regular sampling)
*/
VAR ConstLim; /*YES for constant limit*/

} TYPEOPT;

int OPT(Get)(int user, Planning *pt_plan, Option *opt, Model *mod);
int OPT(FGet)(char **InputFile, int user, Planning *pt_plan, Option *opt, Model *mod);
int OPT(Show)(int user, Planning *pt_plan, Option *opt, Model *mod);
int OPT(Check)(int user, Planning *pt_plan, Option *opt);

#endif

```