

[Help](#)

```
#include "
href../common/optype_h_src.pdfoptype.h"
#include "
href../common/var_h_src.pdfvar.h"
#include "
href../common/ftools_h_src.pdftools.h"
#include "pnl/pnl_random.h"
#include "
href../common/error_msg_h_src.pdferror_msg.h"

/**
 * Get_option:
 * @param user:
 * @param pt_plan:
 * @param option:
 *
 * generic function to interactively read the option
 * parameters
 */
int Get_option_gen(int user, Planning *pt_plan, Option *opt, Model *mod)
{
    int nvar;
    void *pt = (opt->TypeOpt);
    VAR *var = ((VAR *) pt);
    int i;

    (opt->Init)(opt, mod);
    nvar = opt->nvar;
    if (user == TOSCREEN)
        if ((opt->Show)(user, pt_plan, opt, mod))
            do
            {
                Fprintf(TOSCREEN, "_____Option:%s\ n", opt->Na

                for (i = 0; i < nvar; i++)
                {
                    ScanVar(pt_plan, user, &(var[i]));
                }
            }
        }
    }
```

```

        if (var[i].setter) var[i].setter(opt->TypeOpt);
    }
}
while ((opt->Show)(user, pt_plan, opt, mod));

return (opt->Show)(TOSCREENANDFILE, pt_plan, opt, mod);
}

/**
 * FGet_option
 * @param InputFile:
 * @param user:
 * @param pt_plan:
 * @param option:
 * @param model:
 *
 * generic function to read the model parameters from an
 * input file
 */
int FGet_option_gen(char **InputFile, int user, Planning *pt_plan, Option *opt,
{
    int nvar;
    void *pt = (opt->TypeOpt);
    VAR *var = ((VAR *) pt);
    int i;

    (opt->Init)(opt, mod);
    nvar = opt->nvar;
    if (user == TOSCREEN)
        if ((opt->Show)(user, pt_plan, opt, mod))
        {
            Fprintf(TOSCREEN, "_____Option:%s\ n", opt->Name

            for (i = 0; i < nvar; i++)
            {
                FScanVar(InputFile, pt_plan, user, &(var[i]));
                if (var[i].setter) var[i].setter(opt->TypeOpt);
            }
        }
    return (opt->Show)(TOSCREENANDFILE, pt_plan, opt, mod);
}

```

```
}
```

```
/**
 * Generic function to replace the Show member function of
 * the option structures
 *
 * @param user : an integer TOSCREEN or TOFILE
 * @param pt_plan : pointer the planning structure
 * describing what to do
 * @param model : pointer to the model instance
 */
int Show_option_gen(int user, Planning *pt_plan, Option *opt, Model *model)
{
    void *pt = (opt->TypeOpt);
    VAR *var = ((VAR *)pt);
    int nvar = opt->nvar, i;

    Fprintf(user, "##Option:%s\ n", opt->Name);

    for (i = 0; i < nvar; i++)
    {
        PrintVar(pt_plan, user, &(var[i]));
    }
    return (opt->Check)(user, pt_plan, opt);
}
```