

[Help](#)

```
#include "
href.././mod/nonpar1d/nonpar1d_h_src.pdfnonpar1d.h"
#include "
href.././common/chk_h_src.pdfchk.h"
#include "
href.././common/error_msg_h_src.pdferror_msg.h"
#include "
href.././mod/hes1d/hes1d_pad/model_h_src.pdfmodel.h"
#include "premia_obj.h"

static int MOD(Init)(Model *model)
{
    TYPEMOD *pt = (TYPEMOD *) (model->TypeModel);

    if (model->init == 0)
    {
        model->init = 1;
        model->nvar = 0;

        pt->S0.Vname = "Spot";
        pt->S0.Vtype = PDOUBLE;
        pt->S0.Val.V_PDOUBLE = 100.;
        pt->S0.Viter = ALLOW;
        model->nvar++;

        pt->S0_volindex.Vname = "Spot (for Volatility Index Product)";
        pt->S0_volindex.Vtype = PDOUBLE;
        pt->S0_volindex.Val.V_PDOUBLE = 8276.43;
        pt->S0_volindex.Viter = ALLOW;
        model->nvar++;

        pt->R.Vname = "Annual Interest Rate";
        pt->R.Vtype = DOUBLE;
        pt->R.Val.V_DOUBLE = 10.;
        pt->R.Viter = ALLOW;
        model->nvar++;

        pt->implied_volatility.Vname = "Implied Volatility";
        pt->implied_volatility.Vtype = FILENAME;
```

```

    pt->implied_volatility.Val.V_INT = 0;
    pt->implied_volatility.Viter = FORBID;
    pt->implied_volatility.Vsetable = SETABLE;

    model->nvar++;
    if ((pt->implied_volatility.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH_LEN))
        return MEMORY_ALLOCATION_FAILURE;
    sprintf(pt->implied_volatility.Val.V_FILENAME, "%s%simplied_volatility.dat",
        pt->option_prices.Vname = "Volatility Index (for Volatility Index Product)";
    pt->option_prices.Vtype = FILENAME;
    pt->option_prices.Val.V_INT = 0;
    pt->option_prices.Viter = FORBID;
    pt->option_prices.Vsetable = SETABLE;

    model->nvar++;
    if ((pt->option_prices.Val.V_FILENAME = malloc(sizeof(char) * MAX_PATH_LEN))
        return MEMORY_ALLOCATION_FAILURE;
    sprintf(pt->option_prices.Val.V_FILENAME, "%s%soption_prices.dat", premia_

    }
    return OK;
}
TYPEMOD NonPar1d;
MAKEMOD(NonPar1d);

```