

Help

```

#include "bs2d.h"

extern char **error_msg;
extern char* path_sep;

int MOD(Get)(int user,Planning* pt_plan,Model *
    model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);

    MOD(Init)(model);

    if (user==TOSCREEN)
        if ((model->Show)(user,pt_plan,model))
            do
            {
                Fprintf(TOSCREEN,"-----
                -----Model:%s{n",model->Name);

                ScanVar(pt_plan,user,&(pt->T));
                ScanVar(pt_plan,user,&(pt->S01));
                ScanVar(pt_plan,user,&(pt->Mu1));
                ScanVar(pt_plan,user,&(pt->Sigma1));
                ScanVar(pt_plan,user,&(pt->Divid1));
                ScanVar(pt_plan,user,&(pt->S02));
                ScanVar(pt_plan,user,&(pt->Mu2));
                ScanVar(pt_plan,user,&(pt->Sigma2));
                ScanVar(pt_plan,user,&(pt->Divid2));
                ScanVar(pt_plan,user,&(pt->Rho));
                ScanVar(pt_plan,user,&(pt->R));

            }
            while ((model->Show)(user,pt_plan,model)
        );

    return ((model->Show)(TOSCREENANDFILE,pt_plan,
        model));
}

int MOD(Show)(int user,Planning* pt_plan,Model *

```

```

    model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
    VAR r_bs,divid1_bs,divid2_bs;

    Fprintf(user,"##Model:%s{n",model->Name);

    PrintVar(pt_plan,user,&(pt->T));
    PrintVar(pt_plan,user,&(pt->S01));
    PrintVar(pt_plan,user,&(pt->Mu1));
    PrintVar(pt_plan,user,&(pt->Sigma1));
    PrintVar(pt_plan,user,&(pt->Divid1));
    divid1_bs.Val.V_DOUBLE=log(1+pt->Divid1.Val.V_
        DOUBLE/100);
    divid1_bs.Vtype=DOUBLE;
    strcpy(divid1_bs.Vname,"-->Instantaneous Divid
        end Rate 1");
    divid1_bs.Viter=FORBID;
    PrintVar(pt_plan,user,&divid1_bs);
    PrintVar(pt_plan,user,&(pt->S02));
    PrintVar(pt_plan,user,&(pt->Mu2));
    PrintVar(pt_plan,user,&(pt->Sigma2));
    PrintVar(pt_plan,user,&(pt->Divid2));
    divid2_bs.Val.V_DOUBLE=log(1+pt->Divid2.Val.V_
        DOUBLE/100);
    divid2_bs.Vtype=DOUBLE;
    strcpy(divid2_bs.Vname,"-->Instantaneous Divid
        end Rate 2");
    divid2_bs.Viter=FORBID;
    PrintVar(pt_plan,user,&divid2_bs);

    PrintVar(pt_plan,user,&(pt->Rho));
    PrintVar(pt_plan,user,&(pt->R));
    r_bs.Val.V_DOUBLE=log(1+pt->R.Val.V_DOUBLE/100
        );
    r_bs.Vtype=DOUBLE;
    strcpy(r_bs.Vname,"-->Instantaneous Interest
        Rate");
    r_bs.Viter=FORBID;
    PrintVar(pt_plan,user,&r_bs);

```

```

    return (model->Check)(user,pt_plan,model);
}

int MOD(Check)(int user,Planning* pt_plan,Model *
    model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
    int status=OK;
    char helpfile[MAX_PATH_LEN]="";

    if ((2*strlen(model->ID)+strlen("{mod{") +
        strlen("{")
        +strlen("_doc.pdf"))>=MAX_PATH_LEN)
    {
        Fprintf(TOSCREEN,"%s\n",error_msg[PATH_TOO_
            LONG]);
        exit(WRONG);
    }

    strcpy(helpfile,path_sep);
    strcat(helpfile,"mod");
    strcat(helpfile,path_sep);

    strcat(helpfile,model->ID);
    strcat(helpfile,path_sep);

    strcat(helpfile,model->ID);
    strcat(helpfile,"_doc.pdf");

    status+=ChkVar(pt_plan,&(pt->T));
    status+=ChkVar(pt_plan,&(pt->S01));
    status+=ChkVar(pt_plan,&(pt->Mu1));
    status+=ChkVar(pt_plan,&(pt->Sigma1));
    status+=ChkVar(pt_plan,&(pt->Divid1));

    status+=ChkVar(pt_plan,&(pt->S02));
    status+=ChkVar(pt_plan,&(pt->Mu2));
    status+=ChkVar(pt_plan,&(pt->Sigma2));
    status+=ChkVar(pt_plan,&(pt->Divid2));

```

```
status+=ChkVar(pt_plan,&(pt->Rho));
status+=ChkVar(pt_plan,&(pt->R));

return Valid(user,status,helpfile);
}

int MOD(Init)(Model *model)
{
    TYPEMOD* pt=(TYPEMOD*)(model->TypeModel);
    static int first=1;

    if (first)
    {
        strcpy(pt->T.Vname,"Current Date");
        pt->T.Vtype=DATE;
        pt->T.Val.V_DATE=0.;
        pt->T.Viter=ALLOW;

        strcpy(pt->S01.Vname,"Spot 1");
        pt->S01.Vtype=PDOUBLE;
        pt->S01.Val.V_PDOUBLE=100.;
        pt->S01.Viter=ALLOW;

        strcpy(pt->Mu1.Vname,"Trend 1");
        pt->Mu1.Vtype=DOUBLE;
        pt->Mu1.Val.V_DOUBLE=0.;
        pt->Mu1.Viter=ALLOW;

        strcpy(pt->Sigma1.Vname,"Volatility 1");
        pt->Sigma1.Vtype=PDOUBLE;
        pt->Sigma1.Val.V_PDOUBLE=0.2;
        pt->Sigma1.Viter=ALLOW;

        strcpy(pt->Divid1.Vname,"Annual Dividend
Rate 1");
        pt->Divid1.Vtype=DOUBLE;
        pt->Divid1.Val.V_DOUBLE=0.;
        pt->Divid1.Viter=ALLOW;
```

```

    strcpy(pt->S02.Vname,"Spot 2");
    pt->S02.Vtype=PDOUBLE;
    pt->S02.Val.V_PDOUBLE=100.;
    pt->S02.Viter=ALLOW;

    strcpy(pt->Mu2.Vname,"Trend 2");
    pt->Mu2.Vtype=DOUBLE;
    pt->Mu2.Val.V_DOUBLE=0.;
    pt->Mu2.Viter=ALLOW;

    strcpy(pt->Sigma2.Vname,"Volatility 2");
    pt->Sigma2.Vtype=PDOUBLE;
    pt->Sigma2.Val.V_PDOUBLE=0.2;
    pt->Sigma2.Viter=ALLOW;

    strcpy(pt->Divid2.Vname,"Annual Dividend
Rate 2");
    pt->Divid2.Vtype=DOUBLE;
    pt->Divid2.Val.V_DOUBLE=0.;
    pt->Divid2.Viter=ALLOW;

    strcpy(pt->Rho.Vname,"Correlation");
    pt->Rho.Vtype=RGDOUBLEM11;
    pt->Rho.Val.V_RGDOUBLEM11=0.;
    pt->Rho.Viter=ALLOW;

    strcpy(pt->R.Vname,"Annual Interest Rate");
    pt->R.Vtype=DOUBLE;
    pt->R.Val.V_DOUBLE=5.0;
    pt->R.Viter=ALLOW;

    first=0;
}

return OK;
}

TYPEMOD BlackScholes2dim;

MAKEMOD(BlackScholes2dim);

```

References