```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "UnitFormMain.h"
#include "UnitUtils.h"
#include "UnitDMClients.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TformMain *formMain;
//---------------------------------------------------------------------------
__fastcall TformMain::TformMain(TComponent* Owner)
    : TForm(Owner)
{
    // Get Application Version
    strVersion = GetTSLSVersion();
    Caption = Caption + " [ver." + strVersion + "]";
#if defined(__DYNAMIC__)
    Caption += "[Dynamic]";
#endif
#if defined(__LAMBDA_THREAD__)
    Caption += "[Lambda]";
#endif
}
//---------------------------------------------------------------------------
void __fastcall TformMain::FormShow(TObject *Sender)
{
    tsp::Parameters p = dmClients->GetParameters();

    Left = p.intLeft;
    Top  = p.intTop;

    // При подравняване Alignment = alClient на gbLog и memoLog
    // височината Height на memoLog се задава косвено
    // чрез промяна височината Height на цялата форма formMain
    int intLogHeightCompensation = p.intLogHeight - memoLog->Height;
    Height += intLogHeightCompensation;

    if(p.boolLogAuto)
    {
        if(!p.boolLogVisible)
        {
            labelShowHideLogClick(Sender);
        }
    }
    else
    {
        labelShowHideLogClick(Sender);
    }

    if(memoLog->Visible)
    {   // Hide Log Caption and Hint
        labelShowHideLog->Caption = strShowHideLogCaptionHide;
        labelShowHideLog->Hint = strShowHideLogHintHide;
    }
    else
    {   // Show Log Caption and Hint
        labelShowHideLog->Caption = strShowHideLogCaptionShow;
        labelShowHideLog->Hint = strShowHideLogHintShow;
    }

    buttonSync->Enabled = true;
    buttonSync->SetFocus();
}
//---------------------------------------------------------------------------
void __fastcall TformMain::FormClose(TObject *Sender, TCloseAction &Action)
{
    if(dmClients->IsSyncing())
    {
    //  Защита срещу затваряне на приложението при наличие на необслужени заявки.
    //  За да се стигне до тази точка, обработката трябва да бъде по-продължителна (~ sec).
        Action = caNone;
```

```
        String str = "Cannot close while sync [Pending clients #" + IntToStr(dmClients->GetPending(
        MessageDlg(str, mtError, TMsgDlgButtons() << mbOK, 0);
        return;
    }
}
//---------------------------------------------------------------------------
void __fastcall TformMain::buttonSyncClick(TObject *Sender)
{
    // Send Batch of Requests
    dmClients->SendBatchOfReq();
}
//---------------------------------------------------------------------------
void __fastcall TformMain::labelShowHideLogClick(TObject *Sender)
{
    if(memoLog->Visible)
    {   // Hide Log
        labelShowHideLog->Caption = strShowHideLogCaptionShow;
        labelShowHideLog->Hint = strShowHideLogHintShow;

        memoLog->Hide();
        ClientHeight = ClientHeight - memoLog->Height;
    }
    else
    {   // Show Log
        labelShowHideLog->Caption = strShowHideLogCaptionHide;
        labelShowHideLog->Hint = strShowHideLogHintHide;

        ClientHeight = ClientHeight + memoLog->Height;
        memoLog->Show();
    }
}
//---------------------------------------------------------------------------
void __fastcall TformMain::miClearLogClick(TObject *Sender)
{
    int ExitCode = Application->MessageBox(L"Do you want to continue?", L"The System Log will be Cl
        MB_YESNO|MB_ICONWARNING);
    if(ExitCode == IDNO)
        return;

    memoLog->Clear();
    buttonSync->SetFocus();
}
//---------------------------------------------------------------------------
void __fastcall TformMain::miSaveLogClick(TObject *Sender)
{
    String strLogFileName = strLogFilePrefix + "." +
                            FormatDateTime("yyyymmdd.hhmmss", dmClients->GetTSyn()) +
                            strLogFileExpension;
    try
    {
        memoLog->Lines->SaveToFile(strLogFileName);
        MessageDlg(strLogFileName + " saved", mtInformation, TMsgDlgButtons() << mbOK, 0);
    }
    catch(Exception& e)
    {
        MessageDlg(e.Message, mtError, TMsgDlgButtons() << mbOK, 0);
    }
}
//---------------------------------------------------------------------------
void __fastcall TformMain::miSaveReportClick(TObject *Sender)
{
    try
    {
        dmClients->Save();
    }
    catch(Exception& e)
    {
        MessageDlg(e.Message, mtError, TMsgDlgButtons() << mbOK, 0);
    }
}
//---------------------------------------------------------------------------
```