

```

/
#endifdef UnitDMClientsH
#define UnitDMClientsH
//-----
#include <System.Classes.hpp>
#include <System.Win.ScktComp.hpp>
#include <IniFiles.hpp>
//-----
#include <atomic>
#include <vector>
//-----
// ПРОСТРАНСТВО НА ИМЕТА НА TSP (Time Server Protocol)
namespace tsp
{
    // TSP КЛИЕНТ
    class Client
    {
    public:
        enum STATE {Closed = 0, Transient, Open} State; // СЪСТОЯНИЕ
        TClientSocket* csOut; // ИЗХОДЕН КАНАЛ НА КЛИЕНТА

        int intLocalPort;
        TTime timeREQ;
        TTime timeCNC;
        TTime timeDSC;
        TTime timeRPL;
        TTime timeERR;
        unsigned long rpl;
        void __fastcall Init(void);
    };

    // ПАКЕТ TSP КЛИЕНТИ
    class ClientsPack
    {
    public:
        std::vector<Client*> vClients; // КЛИЕНТИ В ПАКЕТА
        std::atomic<int> intPending; // НЕОБСЛУЖЕНИ КЛИЕНТИ
        std::atomic<bool> boolSyncing; // ОБОБЩЕНО СЪСТОЯНИЕ НА ПАКЕТА КЛИЕНТИ

        // ПАРАМЕТРИ //////////////////////////////////////
        //
        String strIPAddress;
        int intVolume;
        bool boolConsecutive;
        int intDelayToFree;
        //////////////////////////////////////

        TDateTime dtStart;
        TDateTime dtEnd;

        void __fastcall Init(void);
        String __fastcall ToMilliSeconds(TTime t, bool boolFormat = false);
        void __fastcall Save(String strFileName);
    };

    // БУФЕРЕН LOG В ПАМЕТА
    class Log
    {
    private:
        std::vector<String> Lines;

    public:
        __fastcall Log(void) {}
        void __fastcall Clear(void) { Lines.clear(); }
        void __fastcall Add(String str);
        void __fastcall Add(TCustomWinSocket* sock, String str);
        void __fastcall Show(TMemo* dst);
    };
}
//-----
class TdmClients : public TDataModule
{

```

```

published:      // IDE-managed Components
TClientSocket *csOut;
void __fastcall csOutConnect(TObject *Sender, TCustomWinSocket *Socket);
void __fastcall csOutDisconnect(TObject *Sender, TCustomWinSocket *Socket);
void __fastcall csOutError(TObject *Sender, TCustomWinSocket *Socket, TErrorEvent ErrorEvent,
    int &ErrorCode);
void __fastcall csOutRead(TObject *Sender, TCustomWinSocket *Socket);

private:      // User declarations
// Конфигурационен файл //////////////////////////////////////
TIniFile* iniFile;
String strIniFileName;

// Променливи на състоянието //////////////////////////////////
tsp::ClientsPack clientsPack;    // пакет TS клиенти
tsp::Log log;                    // системен протокол в паметта

void __fastcall __read_ini_file(void);
void __fastcall __write_ini_file(void);
int __fastcall GetChannelId(TObject *Sender);

public:      // User declarations
__fastcall TdmClients(TComponent* Owner);
__fastcall ~TdmClients(void);

bool __fastcall IsSyncing(void)      { return clientsPack.boolSyncing; }
int __fastcall GetPending(void)      { return clientsPack.intPending; }

void __fastcall SendBatchOfReq(void);    // формиране пакет от заявки
void __fastcall Save(String strFileName)    { clientsPack.Save(strFileName); }
};
//-----
extern PACKAGE TdmClients *dmClients;
//-----
#endif

```