

# InitParams

## Štruktúra InitParams

```
typedef struct _InitParams
{
    unsigned int VerHi; // Verzia DLL rozhrania High DWORD
    unsigned int VerLo; // Verzia DLL rozhrania Low DWORD
    // callback procedury
    void (_stdcall *GetPoint)(HOBj StID, char *Adr, PointParPtr *PtParPtrPtr);
    void (_stdcall *GetFastPoint)(HOBj StID, unsigned int FastAdr, PointParPtr *PtParPtrPtr);
    void (_stdcall *GetStation)(HOBj LnID, char *Adr, StationParPtr *StParPtrPtr);
    void (_stdcall *PointNewValue)(HOBj PtID, KomUniVal *Val, unsigned int Stat);
    void (_stdcall *PointOldValue)(HOBj PtID, KomUniVal *Val, unsigned int Stat, int TransID);
    KomUniVal(_stdcall *PointLastValidValue)(HOBj PtID);
    void (_stdcall *SetStationStatus)(HOBj StID, TStVal Stat);
    unsigned int (_stdcall *WriteToLine)(HOBj StID, void *Data, unsigned int Len);
    unsigned int (_stdcall *ReadFromLine)(HOBj StID, void *Data, unsigned int MaxData);
    void (_stdcall *SetReceiveState)(HOBj StID);
    void (_stdcall *SetTransmitState)(HOBj StID);
    void (_stdcall *ChangeLineMode)(HOBj LnID, unsigned int LineMode);
    void (_stdcall *TraceBufferHex)(HOBj LnID, void *Data, unsigned int Len, TColor *Color);
    void (_stdcall *TraceBufferHexFile)(HOBj LnID, char *FileName, unsigned int MaxFileSize, void *Data,
    unsigned int Len);
    void (_stdcall *TraceBufferASCII)(HOBj LnID, void *Data, unsigned int Len, TColor *Color);
    void (_stdcall *TraceBufferASCIIFile)(HOBj LnID, char *FileName, unsigned int MaxFileSize, void *Data,
    unsigned int Len);
    void (_stdcall *TraceComText)(HOBj LnID, TColor *Color, char *Str);
    void (_stdcall *TraceComTextFile)(HOBj LnID, char *FileName, unsigned int MaxFileSize, char *Str);
    void (_stdcall *TraceComError)(HOBj LnID, char *Str);
    void (_stdcall *TraceComErrorFile)(HOBj LnID, char *FileName, unsigned int MaxFileSize, char *Str);
    void (_stdcall *PointsApply)(HOBj StID, void *AplPar);
    void (_stdcall *StationsApply)(HOBj LnID, unsigned int Prot, void *AplPar);
    void (_stdcall *MessageBox)(char *Title, char *Message);
    unsigned int (_stdcall *CreateThread)(void *Proc, void *Par);
    void (_stdcall *DeleteThread)(unsigned int ThrID);
    void * (_stdcall *GetMem)(unsigned int Len);
    void (_stdcall *FreeMem)(void *Adr);
    D2Time (_stdcall *ActTime)(void);
    D2Time (_stdcall *TimeFromItems)(unsigned int Year, unsigned int Month, unsigned int Day, unsigned int Hour,
    unsigned int Min, unsigned int Sec, unsigned int Msec);
    Bool32 (_stdcall *TimeToItems)(D2Time Tm, unsigned int *Year, unsigned int *Month, unsigned int *Day,
    unsigned int *Hour, unsigned int *Min, unsigned int *Sec, unsigned int *Msec);
    LinListH (_stdcall *CreateList)(void);
    void (_stdcall *DeleteList)(LinListH List);
    void (_stdcall *InsertEl)(LinListH List, void *DataPtr, void *SortProc);
    void (_stdcall *InsertElPrio)(LinListH List, void *DataPtr, unsigned int HighPriority);
    void * (_stdcall *FindEl)(LinListH List, void *DataPtr, void *FindProc);
    void (_stdcall *DeleteEl)(LinListH List, void *DataPtr, void *FindProc);
    void * (_stdcall *FindFirstElAndDelete)(LinListH List);
    void * (_stdcall *FindFirstEl)(LinListH List);
    void * (_stdcall *FindNextEl)(LinListH List);
    void * (_stdcall *FindPreviousEl)(LinListH List);
    unsigned int (_stdcall *GetNumber)(LinListH List);
    void (_stdcall *StPerf_CommActivity)(HOBj StID);
    void (_stdcall *StPerf_TimeSynchro)(HOBj StID);
    void (_stdcall *StPerf_IncFramesIn)(HOBj StID);
    void (_stdcall *StPerf_IncFramesOut)(HOBj StID);
    void (_stdcall *StPerf_IncRetryErr)(HOBj StID);
    void (_stdcall *LnPerf_IncTxBytes)(HOBj LnID, unsigned int Increment);
    void (_stdcall *LnPerf_IncRxBytes)(HOBj LnID, unsigned int Increment);
    void (_stdcall *LnPerf_IncRetryErr)(HOBj LnID);
    void (_stdcall *LnPerf_IncFramesIn)(HOBj LnID);
    void (_stdcall *LnPerf_IncFramesOut)(HOBj LnID);
    unsigned int Prot; // Komunikacny protokol
    void (_stdcall *GetRdStateAndInstance)(Bool32 *RdHot, Bool32 *Instance);
    void (_stdcall *GetOemProtName)(unsigned int spProtocol, char *ProtName);
} InitParams;
```

Štruktúra *InitParams* je prenesená do protokolovej DLL povinným volaním [Init](#). Obsahuje číslo verzie protokolového rozhrania a adresy [call-back funkcií](#).



**Súvisiace stránky:**

[D2000 KomAPI - štruktúry](#)