

Štruktúry rozhrania (D2000 OBJApi)

D2000 OBJApi - Štruktúry rozhrania

Od verzie D2000 V9.02.034 je dostupná 32 aj 64-bitová verzia s upraveným hlavikovým súborom pre použitie v projektoch s interpretáciou textových reazcov v 8-bitovej forme (ANSI) alebo v 16-bitovej forme (wide character UNICODE), podľa nastavenia parametra projektu "Character Set" v prostredí MS Visual Studio.

Štruktúra UniVal

Štruktúra *UniVal* obsahuje všetky informácie o hodnote a stave objektu systému D2000.

Deklarácia v jazyku C:

```
typedef struct _UniVal
{
    GenValueType gvaltyp;
    unsigned int status;
    tLimitStatus limitStatus;
    tProcAlarmType procAlarmStatus;
    ValueType type;
    double valtime;
    double procAlarmTime;
    unsigned int Flags;
    VOBJ Indirect;
    union
    {
        tBVal Boval;
        int Intval;
        double Realval;
        StVal Stval;
        struct
        {
            {
                AlVal Alval;
                double AlTimes[4];
            } a;
            PrVal Prval;
            double TmAval;
            double TmRval;
            tQValue QVal;
            char *TxtVal;
            ArrayRecPtr ArrayValPtr;
            struct
            {
                ArrayRecPtr recordValPtr;
                HOBJ StructTypId;
                int ColsNr;
            } r;
        } v;
    } UniVal;
```

Popis jednotlivých astí štruktúry:

- **gvaltyp**
Určuje typ hodnoty objektu pre variantnú as štruktúry *UniVal*.
- **status**
Stav hodnoty, nadobúda kombináciu hodnôt:

Definícia konštanty	Popis
#define VAL_Valid 1U	Hodnota objektu je platná.
#define VAL_ProcAlarm 2U	Objekt má aktívny procesný alarm.
#define VAL_NoAckPAlarm 4U	Objekt mal aktívny procesný alarm, ktorý nebol potvrdený.
#define VAL_PrAISilent 8U	Procesné alarmy objektu sa nevyhodnocujú.
#define VAL_Weak 16U	Hodnota objektu je podozrivá – "weak".

#define VAL_NoAckValue 32U	Objekt zmenil hodnotu a táto nebola potvrdená.
#define VAL_Transient 64U	Hodnota objektu je v prechodnom stave pri zápise.
#define VAL_Default 128U	Hodnota je v implicitnom stave.
#define VAL_Manual 256U	Hodnota objektu bola zmenená rune.
#define VAL_PrAlCrit 512U	Objekt má aktívny kritický procesný alarm.

- **limitStatus**

Stav hodnoty objektu vzhľadom ku konfigurovaným medziam hodnoty:

```
typedef enum {InLimit, VL_Limit, L_Limit, H_Limit, VH_Limit, LimitsProblem} tLimitStatus;
```

Stav hodnoty	Popis stavu
InLimit	Hodnota objektu je v rozsahu platných medzí.
L_Limit	Hodnota objektu klesla pod dolnú medzu.
VL_Limit	Hodnota objektu klesla pod najnižšiu medzu.
H_Limit	Hodnota objektu prekroila hornú medzu.
VH_Limit	Hodnota objektu prekroila najvyššiu medzu.
LimitsProblem	Objekt má problém s vyhodnotením medzí (neznáma hodnota aktívnej medze, aktívne medze prekročili svoje hodnoty).

- **procAlarmStatus**

Typ aktívneho procesného alarmu.

```
typedef enum {NoAlarm, ToOn, ToOff, On, Off, Err, Oscillate, ErrCmdOn, ErrCmdOff, SwToTrans, SwToOff, SwToOn, SwToErr, SwTrans, SwOff, SwOn, SwErr, ErrZa1CmdOff, HL, VHL, LL, VLL, ToHL, ToVHL, ToLL, ToVLL, ErrWriteCmd, A28, A29, A30, A31, A32} tProcAlarmType;
```

- **type**

Typ objektu systému D2000.

- **valtime**

as poslednej zmeny hodnoty objektu.

- **procAlarmTime**

as poslednej zmeny procesného alarmu objektu.

- **Flags**

Bitové pole: Príznaky hodnoty objektu (A až P).

- **Indirect**

Ak hodnota pochádza z položky [štruktúry](#) typu *Object*, je tu zdroj hodnoty (*id* objektu a *row*, *col*).

Variantná as, nesúca vlastnú hodnotu objektu:

- **Boval**

Hodnota objektu typu *Boolean* (type : Bo, Di, Do, De, Li)

- **Intval**

Hodnota objektu celočíselného typu (type : Int, Ci, Co, Ce)

- **Realval**

Hodnota objektu reálno-číselného typu (type : Re, Ai, Ao, Ae)

- **Stval**

Hodnota objektu typu stanica (type : St)

- **a.Alval**

Hodnota objektu typu alarm (type : Al)

- **a.ALTimes**

asy jednotlivých [stavov](#) alarmov indexované:

Stav alarmu	Index
Alarm	0
NoKvit	1
Kvit	2
Norm	3

!!! UPOZORNENIE !!!

Od verzie 7.0 sa položka *a.AITimes* nepoužíva a preto bola premenovaná na *a.AITimes_Unused*. Prvky poa sú plnené hodnotou 0.0.

- **Prval**
Hodnota objektu typu proces (type : Pr)
- **TmAVal**
Hodnota objektu absolútnych asových typov (type : TmA, TiA, ToA).
- **TmRval**
Hodnota objektu relatívnych asových typov (type : TmR, TiR, ToR).
- **Qval**
Hodnota štvorstavového vstupu (type : Qi).
- **TxtVal**
Hodnota objektu typu text (type : Txt, TxtI, TxtO).
- **ArrayValPtr**
Hodnota objektu typu pole (type : Arr).
- **r.recordValPtr**
Odkaz na hodnotu štruktúry (type : Rec).
- **r.StructTypId**
Id objektu typu *Definícia štruktúry*, ktorá definuje typ hodnoty.
- **r.ColsNr**
Počet stpcov v štruktúre.

Upozornenie pre niektoré typy hodnôt:

Štruktúra *UniVal* obsahuje pre objekty typu **text** a **pole** iba odkazy na hodnoty. Vlastné hodnoty sú vytvorené dynamicky a treba ich po použití uvoni volaním funkcie [FreeData](#). Výnimku tvorí asynchrónne volanie obsluhy novej hodnoty [NewValueProc](#) (po návrate z obsluhy je uvonenie pamäti hodnôt vykonané v réžií ObjAPI) a funkcia [ListOfObjects](#) (hodnoty sú uvonené pri uvonení štruktúr typu *ListObjData* volaním funkcie [FreeData](#)).

Reprezentácia polí a štruktúr:

Polia a štruktúry sú reprezentované poom hodnôt typu *UniVal* pred ktorým sú dolný (*lowIndex*) a horný (*hiIndex*) index. Položky poa sú indexované od 1. Za položkou *hiIndex* nasledujú postupne jednotlivé hodnoty po riadkoch.

```
typedef struct _ArrayRec
{
    int lowIndex;
    int hiIndex;
} ArrayRec;
typedef ArrayRec * ArrayRecPtr;
```

Prístup k položke štruktúry (riadok, stpec) znázoruje nasledovná funkcia:

```
UniValPtr GetRecordItem(UniVal value, int row, int col)
{
    UniValPtr uni_Ptr;
    int valueIdx;
    valueIdx = (row-1)*value.v.r.ColsNr+col - 1;
    // adresa prvej hodnoty
    uni_Ptr = (UniValPtr)((char *)value.v.r.recordValPtr +
        sizeof(*value.v.r.recordValPtr));
    uni_Ptr = &(uni_Ptr[valueIdx]);
    return uni_Ptr;
}
```

Pri volaní procedúr rozhrania je dôležité, aby boli indexy korektne vyplnené !!!