

Lokálne premenné (Event Script Language (ESL))

Lokálne premenné v skripte

Pomocou lokálnych premenných je možné v skriptoch efektívne implementovať cykly, podprogramy a iné riadiace algoritmy.

Identifikátor lokálnej premennej je postupnosť znakov, ktoré sú prípustné pre meno objektu v systéme D2000 ([pravidlá zápisu mien objektov](#)). Musí sa začínať znakom "_" (podčiarkovník). Napríklad: `_i`, `_i1`, `_locVar.1`. Maximálny počet znakov je 64. Hodnota lokálnej premennej sa vyznačuje všetkými atribútmi hodnoty, ktoré majú objekty v systéme D2000 (as vznik, limity, procesný alarm, at.).

Dostupné typy lokálnych premenných:

- **INT** - celočíselná premenná v rozsahu -2147483647 .. 2147483648
- **BOOL** - logická premenná (boolean)
- **REAL** - reálna premenná v rozsahu -1.7E+308 .. 1.7E+308
- **TIME** - premenná typu Absolútny čas
- **TEXT** - premenná typu Text
- **ALIAS** - odkaz na objekt systému D2000.
- **RECORD** - štruktúrovaná hodnota

Poznámka

Pri demonštrácii niektorých vlastností lokálnych premenných sú použité objekty [SD.RecordDef](#) a [SV.Struktura](#).

Deklarácia lokálnej premennej je na riadku uvedená jej typom (pri typoch RECORD a ALIAS typom štruktúry). Za ním nasledujú mená deklarovaných lokálnych premenných oddelené iarkou, napr.

`INT _a, _b, _c,`
alebo

`RECORD NOALIAS (SD.RecordDef) _rec1, _rec2.`

TIME je aj kúové slovo v [priradení](#).

Premenné typu **INT**, **BOOL**, **REAL**, **TIME** a **TEXT** predstavujú jednoduché hodnoty daného typu.

Premenné typu **TEXT** sú interne kódované v [UTF-8](#).

Premenná typu **ALIAS** predstavuje odkaz na objekt systému D2000. Po štarte skriptu je odkaz prázdny a jeho hodnota je nedefinovaná. Taktiež použitie takejto premennej nie je možné a generuje chybu `_ERR_NO_ASSIGNED_ALIAS`, ktorá má za následok prerušenie exekúcie skriptu. Po jej inicializácii (nastavenie odkazu na objekt) príslušnou akciou ([SET AS](#)) preberá hodnotu objektu, na ktorý je nasmerovaná (asociovaná). Je možné ju použiť vo výrazoch, alebo do nej priradiť hodnotu s tým istým významom ako keby sme použili asociovaný objekt.

Príklad:

```
ALIAS _obj
INT _i
_obj := 1 ; akcia generuje chybu ERR_NO_ASSIGNED_ALIAS
```

```
ALIAS _obj
INT _i
SET _obj AS U.Int
_obj := 1 ; priradenie hodnoty 1 do lokálnej premennej U.Int
WAIT ; akcie na realizáciu priradenia
_i := _obj ; priradí hodnotu premennej U.Int do lokálnej premennej _i
```

Ak je k lokálnej premennej typu ALIAS asociovaný objekt typu [Pole hodnôt](#), je možné používať indexy tak, ako pri objekte typu [Pole hodnôt](#):

Príklad:

```
ALIAS _obj
INT _i
SET _obj AS X.Array
_i := _obj[2] ; priradí hodnotu druhej položky poľa X.array do lokálnej premennej _i
```

Rozlišujú sa dva druhy lokálnych premenných typu **ALIAS**:

- *netypový ALIAS*. Ide o doteraz popisovaný ALIAS
- *typový ALIAS*. Odkaz na objekt typu [Štruktúrovaná premenná](#) určitého typu.

Pri deklarácii typového ALIASu je nutné uviesť typ (v našom prípade objekt typu [Definícia štruktúry](#)), ktorý obmedzuje množinu možných objektov (typu [Štruktúrovaná premenná](#)), s ktorými môže byť lokálna premenná (typový ALIAS) asociovaná. Pri takto deklarovanej lokálnej premennej je možné vo výrazoch (samozrejme po asociovaní akciou [SET AS](#)) používať indexovanie a prístup k položkám tak, ako pri objekte typu [Štruktúrovaná premenná](#).

Príklad:

```
ALIAS (SD.RecordDef) _recAlias
INT _i
```

```
_i := 2
ALIAS _recAlias AS SV.Structure
_recAlias[_i]^Int := 1
```

Premenná typu **RECORD** umožňuje založiť lokálnu premennú typu *Record*. Typ štruktúry je daný objektom typu [Definícia štruktúry](#), ktorý je uvedený pri deklarácii. Ak premenná typu RECORD je formálny parameter procedúry, nemusí byť zadán jeho typ štruktúry - [netypový RECORD](#).

Príklad:

```
RECORD (SD.RecordDef) _recLocal
```

Po štarte skriptu je rozmer poa takto definovanej lokálnej premennej jedna (1) a všetky položky majú neplatnú hodnotu.

Príklad: Priradenie hodnoty

```
RECORD (SD.RecordDef) _recLocal
_recLocal[1]^Text := "text value"
_recLocal[1]^Int := 8
```

Zmena rozmeru poa lokálnej premennej je možná pomocou akcie [REDIM](#).

Príklad - Zmena rozmeru poa na desa (10) prvkov

```
REDIM _recLocal [10]
```

Minimálny rozmer poa je nula (0). Zmena rozmeru poa nemá za následok stratu hodnôt v položkách, ktoré existovali v poli pred akciou. Prípadné novo vzniknuté položky sú inicializované na neplatnú hodnotu.

Položka *lokálnej premennej* typu *Record* (alebo objektu typu [Štruktúrovaná premenná](#)) môže byť typu *Objekt* (pozri kapitolu [Definícia štruktúry](#)). Takáto položka sa v skripte interpretuje ako *netypový ALIAS*. Dôsledok:

Priradenie

```
_recLocal[1]^Object := 1
```

môže vyvola chybu *_ERR_NO_ASSIGNED_ALIAS* ak k položke nie je asociovaný objekt. Priradenie objektu položke je zhodné s priradením objektu pri premennej typu ALIAS:

```
SET _recLocal[1]^Object AS U.Int
```

Poznámka: Podobne ako pri netypovom ALIASe, prístup na položku lokálnej premennej, ktorá nie je asociovaná má za následok generovanie chyby *_ERR_NO_ASSIGNED_ALIAS*.

Táto vlastnosť sa dá potlačiť pri deklarácii lokálnej premennej kúovým slovom NOALIAS.

```
RECORD NOALIAS (SD.RecordDef) _recLocal
```

Pri takto deklarovanej lokálnej premennej sú všetky položky typu *Objekt* interpretované ako normálne hodnoty ubovoného typu (INT, BOOL, REAL, TEXT, TIME). Je teda korektný nasledovný zápis:

```
_recLocal[1]^Object := 1
_recLocal[1]^Object := "TEXT"
```

Položka *Objekt* s jednotlivými priradeniami mení svoj typ.

Pri priradení hodnoty do položky typu *Objekt* objektu typu [Štruktúrovaná premenná](#) bude priradenie automaticky presmerované do asociovaného objektu (podobne je to pri netypovom ALIASe). Ak položka nemá asociovaný objekt, chyba sa prejaví len pri následnej akcii [WAIT](#).

Poznámka

- alej v texte pojem *Štruktúra* predstavuje hodnotu lokálnej premennej typu *Record* alebo hodnotu objektu typu [Štruktúrovaná premenná](#).

ESL umožňuje priradiť celý jeden riadok štruktúry alebo celú štruktúrou naraz. Toto priradenie je realizované akciou [SET WITH](#).

Syntax akcie:

```
SET identDst_Rec WITH identSrc_Rec
```

priom

- identSrc_Rec - predstavuje zdroj hodnôt,
- identDst_Rec - predstavuje cieľ hodnôt.

Oba identifikátory musia predstavovať riadok *štruktúry* alebo celú *štruktúru*. Typy štruktúr musia byť rovnaké (Chyba: *Nekompatibilné typy*). Ak sa jedná o celú štruktúru, musia mať zhodný počet riadkov (Chyba: *_ERR_RANGE_ERROR*). Akcia priradí postupne všetky hodnoty zo zdroja (*identSrc_Rec*) do cieľa (*identDst_Rec*).

!!! NEMENÍ PRIPOJENÉ (ASOCIOVANÉ) OBJEKTY V JEDNOTLIVÝCH POLOŽKÁCH !!!

Akcia **SET AS** umožňuje priradiť a meniť pripojené objekty (v položkách typu *Objekt*) v celom riadku štruktúry, alebo v celej štruktúre naraz.

Syntax akcie:

```
SET identDst_Rec AS identSrc
```

Pre *identSrc* a *identDst* platia tie isté obmedzenia ako pri akcii **SET WITH**, avšak nesmú to byť lokálne premenné typu *Record* deklarované ako *NOALIAS*.

Prípustné kombinácie parametrov *identSrc* a *identDst* pre akcie **SET WITH a **SET AS**.**

Akcia **SET BIND** asociovuje **riadok** lokálnej premennej typu *Record* s riadkom objektu typu **Štruktúrovaná premenná**.

Syntax akcie:

```
SET _recLocal_Row BIND struct_Row
```

priom

- *_recLocal_Row* - predstavuje odkaz na riadok lokálnej premennej typu *Record*
- *struct_Row* - predstavuje riadok objektu typu **Štruktúrovaná premenná**

Príklad:

```
SET _recLocal[2] BIND SV.Structure[1]
```

Po vykonaní akcie budú hodnoty všetkých položiek riadku 2 lokálnej premennej (*_recLocal[2]*) zhodné s hodnotami položiek riadku 1 objektu *SV.Struktura*. Napríklad odkaz na hodnotu *_recLocal[2]^Int* vráti tú istú hodnotu ako odkaz *SV.Struktura[1]^Int*.

Vlastnosť asociovania riadku je realizovaná do dôsledku.

Príklad: Priradenie hodnoty

```
_recLocal[2]^Int := 1
```

má za následok priradenie hodnoty 1 do položky *SV.Struktura[1]^Int*. Keže hodnota položky *_recLocal[2]^Int* kopíruje hodnotu položky *SV.Struktura[1]^Int*, zmena v lokálnej premennej sa prejaví až po zmene v objekte *SV...* (možné použiť akciu **WAIT**). Riadky lokálnej premennej typu *Record*, ktoré nie sú spojené (**SET BIND**) s iným riadkom majú chovanie nezmenené (popísané v predchádzajúcich odstavcoch).

Každý riadok lokálnej premennej *_recLocal* môže byť spojený s iným (alebo aj s tým istým) riadkom ubovoného objektu typu **Štruktúrovaná premenná**. Jediné obmedzenie je, že lokálna premenná a objekt musia byť rovnakého štruktúrovaného typu (*definícia štruktúry*).

Rozpojenie riadku (zrušenie akcie **SET BIND**) je možné akciou:

```
SET _recLocal[2] BIND NONE
```

Všetky hodnoty položiek sa dostanú do nedefinovaného stavu.



Súvisiace stránky:

[Akcie v skriptoch](#)