

# Prístup k dátam

[Aktuálne hodnoty](#)  
[Historické údaje](#)  
[Volanie ESL RPC procedúr](#)  
[Obmedzenia](#)  
[Význam stpcov so stavovými informáciami](#)

## Aktuálne hodnoty

---

Základné aktuálne hodnoty sú roztriedené podľa typu hodnoty do nasledujúcich piatich tabuliek.

**1. AnalogPoints** - reálne hodnoty a hodnoty s relatívnym asom.

MENO STPCA	TYP HODNOTY	VEKOS
NAME	CHAR	64
DESCRIPT	CHAR	50
VALUE	DOUBLE	8
STATUS	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

**2. IntegerPoints** - celoíselné hodnoty.

MENO STPCA	TYP HODNOTY	VEKOS
NAME	CHAR	64
DESCRIPT	CHAR	50
VALUE	INTEGER	4
STATUS	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

**3. EnumPoints** - vymenované typy vrátane typu *Boolean*.

MENO STPCA	TYP HODNOTY	VEKOS
NAME	CHAR	64
DESCRIPT	CHAR	50
VALUE	INTEGER	4
STATUS	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

**4. TimePoints** - hodnoty s absolútnym asom.

MENO STPCA	TYP HODNOTY	VEKOS
NAME	CHAR	64

DESCRIPT	CHAR	50
VALUE	TIMESTAMP	
STATUS	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

5. TextPoints

MENO STPCA	TYP HODNOTY	VEKOS
NAME	CHAR	64
DESCRIPT	CHAR	50
VALUE	CHAR	254
STATUS	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

Objekty typu [Pole hodnôt](#) a [Štruktúrovaná premenná](#) tvoria samostatné tabuky. Meno tabuky je odvodené od mena premennej nahradením znakov bodka "." znakom podčiarkovník "\_" v mene.

Každá premenná typu [Pole hodnôt](#) tvorí tabuku s nasledovným formátom:

MENO STPCA	TYP HODNOTY	VEKOS
TIME	TIMESTAMP	
VALUE	DOUBLE	8
STATUS	INTEGER	4
ROW_ID	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4
LIMITS	INTEGER	4
FLAGS	INTEGER	4

Každá premenná typu [Štruktúrovaná premenná](#) tvorí tabuku s formátom zodpovedajúcim formátu štruktúrovanej premennej. Tabuka je vždy rozšírená o jeden stpec ROW\_ID urujúci poradové číslo - riadok štruktúrovanej premennej.

Historické údaje

Objekty typu [Archivovaná hodnota](#) tvoria samostatné tabuky. Meno tabuky je odvodené od mena archivovanej premennej nahradením znakov bodka "." znakom podčiariťnik "\_" v mene.

Každá premenná typu [Archivovaná hodnota](#) tvorí tabuku s nasledovným formátom:

MENO STPCA	TYP HODNOTY	VEKOS
TIME	TIMESTAMP	
VALUE	DOUBLE	8
STATUS	INTEGER	4
STEP	INTEGER	4
WEAK	INTEGER	4
DEFAULT	INTEGER	4

LIMITS	INTEGER	4
FLAGS	INTEGER	4
ARCH_FLAGS	INTEGER	4

Implementovaná je tabuľka **ArchivValues**, cez ktorú je možné získať hodnoty viacerých archívnych objektov v rovnakých časových bodoch. Pri výbere z tejto tabuľky je možné zadať viac mien archívnych objektov v jednom príkaze.

Príklad:

```
SELECT TIME,VALUE01,STATUS01,VALUE02,STATUS02 FROM ArchivValues WHERE TIME >{ts '2000-04-11 10:00:00'} and TIME <{ts '2000-04-11 12:00:00'} and NAME01="H.AAA1" and NAME02="H.AAA2" and STEP=60
```

Tabuľka **ArchivValues** má nasledovnú štruktúru:

MENO STPCA	TYP HODNOTY	VEKOS
TIME	TIMESTAMP	
STEP	INTEGER	4
NAME01	CHAR	64
VALUE01	DOUBLE	8
STATUS01	INTEGER	4
WEAK01	INTEGER	4
DEFAULT01	INTEGER	4
LIMITS01	INTEGER	4
FLAGS01	INTEGER	4
ARCH_FLAGS01	INTEGER	4
NAME02	CHAR	64
VALUE02	DOUBLE	8
STATUS02	INTEGER	4
WEAK02	INTEGER	4
DEFAULT02	INTEGER	4
LIMITS02	INTEGER	4
FLAGS02	INTEGER	4
ARCH_FLAGS02	INTEGER	4
...		
NAME <sub>n</sub>	CHAR	64
VALUE <sub>n</sub>	DOUBLE	8
STATUS <sub>n</sub>	INTEGER	4
WEAK <sub>n</sub>	INTEGER	4
DEFAULT <sub>n</sub>	INTEGER	4
LIMITS <sub>n</sub>	INTEGER	4
FLAGS <sub>n</sub>	INTEGER	4
ARCH_FLAGS <sub>n</sub>	INTEGER	4

Kde *n* je maximálne 12.

## Volanie ESL RPC procedúr

Pomocou [D2000 ODBC Driver](#) je umožnené aj volanie ESL RPC procedúr zodpovedajúce ESL akcii [CALL](#). Pre tento účel sú vytvorené špeciálne tabuľky, ktorých mená sú odvodené od mena procesu [D2000 EventHandler](#), na ktorom má byť procedúra vykonaná, objektu typu [event](#) a názvu [RPC procedúry](#). Tieto sú spojené znakom "^" (napr. SELF\_EVH^E\_event^proc), pričom bodky "." v názvoch sú nahradené znakom podčiarkovník "\_". Tabuľky, ktoré v názve neobsahujú názov procesu Event handler (napr. E\_event^proc) slúžia na broadcast volanie RPC procedúr. Stĺpce týchto tabuliek sú pomenované podľa parametrov procedúry. Parametre typu [RECORD](#) sú reprezentované stĺpcami, ktoré sú tvorené kombináciou názvu parametra a názvu stĺpca štruktúry (parameter^stlpec). Rovnako ako tabuľky štruktúrovaných premenných sú doplnené o stĺpec "ROW\_ID". V menách parametrov sú taktiež bodky "." nahradzované znakom podčiarkovník "\_". Tabuľky RPC procedúr sú navyše doplnené o stĺpec "\$async" a "\$instanceId".

Volanie procedúry môže byť:

- synchronónne - v SELECT asti sa nepoužije stpec "\$async",
- asynchronónne - v SELECT asti sa použije jediný stpec "\$async",
- broadcast - použije sa RPC tabuka bez názvu Event handlera.

V prípade adresovania volania procedúry na konkrétnu inštanciu sa vo WHERE asti použite stpec "\$instanceId".

## Všeobecná syntax volania RPC procedúry:

```
SELECT $async | param1, param2, param3, ...
FROM [proces_EVH^]E_event^proc
[WHERE param3 = x AND param4 = y AND ... [$instanceId = id]]
```

## Príklady volania RPC procedúr cez ESL a ich ODBC ekvivalent:

synchronónne volanie:	CALL E.event MyProc(_param1, 2) SYNC ON SELF.EVH SELECT _param1 FROM SELF_EVH^E_event^MyProc WHERE _param2 = 2
asynchronónne volanie:	CALL E.event MyProc(1, 2) ASYNC ON SELF.EVH SELECT \$async FROM SELF_EVH^E_event^MyProc WHERE _param1 = 1 AND _param2 = 2

WHERE as môže obsahovať len priradenia hodnôt parametrom.

Výstup SELECT-u bude mať toľko riadkov, akoko má parameter typu RECORD s najväčším potom riadkov, resp. 1 riadok, ak procedúra má len neštruktúrované typy parametrov. V prípade, že počas vykonávania procedúry došlo k výnimke, exekúcia SQL dotazu skoní neúspešne. Výsledok asynchronónneho volania procedúry nie je možné zistiť, a preto takýto dotaz nevracia žiadne dáta.

## Obmedzenia

### Pre príkaz SELECT platia nasledovné obmedzenia:

1. V príkaze SELECT môže byť uvedená len jedna tabuľka.
2. Triedenie je povolené len podľa hodnoty jedného stpca (ORDER BY).
3. Select archívnej hodnoty musí obsahovať ohraničenie časového intervalu napr.:

```
SELECT TIME,VALUE,STATUS FROM H_NAME WHERE (TIME >"10-04-2000" AND TIME <"12-04-2000 10:00")
```

Voliteľne môže obsahovať definovanie požadovaného časového kroku v sekundách.

```
SELECT TIME,VALUE,STATUS FROM H_NAME WHERE (TIME >"10-04-2000" AND TIME <"12-04-2000 10:00") AND STEP=60.
```

Syntax zadávania absolútneho času "dd-mm-rrrr [hh[:mi[:ss[:mss]]]]" alebo {ts '2000-04-11 10:00:00'}.

### Pre volanie RPC procedúr platia nasledovné obmedzenia:

1. Nie je možné vola RPC procedúry, ktoré:
  - nemajú žiadne parametre,
  - obsahujú parametre typu ALIAS,
  - obsahujú IN parametre typu RECORD.
2. Stpec \$instanceId RPC tabuliek sa môže nachádzať len vo WHERE asti SQL dotazu.
3. Stpec \$async RPC tabuliek sa môže nachádzať len v SELECT asti SQL dotazu a nemôže byť použitý v kombinácii s iným stpcom.
4. Stpce RPC tabuliek, ktoré reprezentujú stpce parametrov typu RECORD, nemôžu byť použité vo WHERE asti SQL dotazu.
5. Stpce RPC tabuliek reprezentujúce parametre typu IN nemôžu byť použité v SELECT asti SQL dotazu.
6. Pri volaní RPC procedúry nie je podporená ORDER BY klauzula ani agregované funkcie (count, avg, max, ...).

## Význam stpcov so stavovými informáciami

<b>STATUS</b>	- hodnota je platná ak STATUS = 1
<b>WEAK</b>	- hodnota je Weak ak WEAK = 1
<b>DEFAULT</b>	- hodnota je v stave Default ak DEFAULT = 1
<b>LIMITS</b>	- stpec môže nadobúdať nasledovné hodnoty:  InLimit = 0 , VL_Limit = 1 , L_Limit = 2 , H_Limit = 3 , VH_Limit = 4 , LimitsProblem = 5
<b>FLAGS</b>	- stpec nadobúda hodnoty Extended flagov (ako bitset):  BF_A = 1, BF_B = 2, BF_C = 4, BF_D = 8, BF_E = 16, BF_F = 32, BF_G = 64, BF_H = 128, BF_I = 256, BF_J = 512,  BF_K = 1024, BF_L = 2048, BF_M = 4096, BF_N = 8192, BF_O = 16384, BF_P = 32768

**ARCH\_FLAGS** - stpec nadobúda hodnoty [archívnych príznakov](#) (ako bitset - suma nasledovných [konštánt](#)).



**Súvisiace stránky:**

[D2000 ODBC Driver](#)

[Konfigurácia DSN \(Data Source Name\)](#)