

# PerfCounterR

## Funkcie %PerfCounter, %PerfCounterR

### Funkcia

Funkcia vráti hodnotu uvedeného "Performance Counter".

### Deklarácia

```
INT %PerfCounter(  
    TEXT in text  
)
```

```
REAL %PerfCounterR(  
    TEXT in text  
)
```

### Parametre

<b>text</b>	Definícia Performance Counter.
-------------	--------------------------------

### Popis

Definícia countra má nasledovnú syntax:

\\ComputerName\Object(instance)\Counter

Ak sa sleduje Counter na lokálnom počítači, prvá časť udávajúca meno počítača sa môže vynechať. Ak má objekt len jednu inštanciu, časť "(instance)" sa vynecháva. Ak je funkcia %PerfCounter (%PerfCounterR) volaná aspoň raz za sekundu, vracia nesprávne hodnoty (0). V prípade potreby volania funkcie každú sekundu pre viac systémových informácií, je riešením vykonať každé volanie na inom riadku v ESL skripte.

### Príklad

```
%PerfCounter("\\D2000 Server\MemUsed") ; vráti hodnotu pamäte  
použitej procesom D2000 Server
```

### Poznámka 1

Niekedy (pre špecifické počítadlá) vracia funkcia %PerfCounter pri prvom volaní vždy nulu - napr. pre počítadlo záťaž procesora: "\\Processor(\_Total)\% User Time". Vtedy je nutné volať túto funkciu dvakrát za sebou v sluke s oneskorením (veľkosť treba nastaviť empiricky) medzi volaniami. Pre počítadla iného charakteru (napr. "Memory\Available MBytes", "LogicalDisk(C:)\% Free Space" alebo "\\Process(\_Total)\Handle count") funguje táto funkcia bez problémov a vracia korektnú hodnotu hneď pri prvom volaní. Zrejme je problém s počítadlami, ktoré vyžadujú uplynutie určitého času (na zber vzorky dát) medzi registrovaním si počítadla a načítaním jeho hodnoty.

Príklad takéhoto kódu, ktorý volá funkciu %PerfCounter dvakrát s oneskorením 0.1 sekundy, pokiaľ pri prvom volaní bola načítaná hodnota 0:

```
INT _i  
REAL _result  
  
FOR _i=1 TO 2 DO_LOOP  
    _result := %PerfCounter("\\Processor(_Total)\% User Time")  
    IF _result<0 THEN  
        IF _result=0 & _i=1 THEN ; workaround - pri prvom volaní (napr.  
na \\Processor (_Total)\% User Time) vracia 0  
            DELAY 0.1 [s]  
        ELSE  
            EXIT_LOOP  
        ENDIF  
    ELSE  
        EXIT_LOOP  
    ENDIF  
END_LOOP
```

Tento kód nefunguje - preto, lebo sa nevolá dvakrát ten istý riadok, tá istá inštancia PerfCounter funkcie:

```
INT _i
REAL _result

_result := %PerfCounter(_counter)
IF _result\VLD THEN
  IF _result=0 THEN          ; NEFUNKCNY workaround - pri prvom volaní (napr.
na \Processor(_Total)\% User Time) vracia 0
    DELAY 0.05 [s]
    _result := %PerfCounter(_counter)
  ENDIF
ENDIF
```

#### Poznámka 2

Funkcia %PerfCounterR je vhodná vtedy, ak je predpoklad, že hodnota bude väčšia ako 2 147 483 647.



#### Súvisiace stránky:

[Implementované funkcie](#)  
[Argumenty funkcie - typy](#)