

# PROCEDURE

## Akcia PROCEDURE

### Deklarácia

```
PROCEDURE ProcName [([IN] type1 paramName1[,paramName2, ...] [IN] type2
paramName3]...)]

; akcie

END ProcName
```

### Deklarácia pre vzdialené volanie

```
[IMPLEMENTATION] RPC PROCEDURE [ESLInterface^]ProcName [([IN] type1
paramName1[,paramName2, ...] [IN] type2 paramName3]...)]

; akcie

END ProcName

[IMPLEMENTATION] RPCX PROCEDURE [ESLInterface^]ProcName [([IN] type1
paramName1[,paramName2, ...] [IN] type2 paramName3]...)]

; akcie

END ProcName
```

### Deklarácia pre volanie do UNITu

```
PUBLIC PROCEDURE ProcName [([IN] type1 paramName1[,paramName2, ...] [IN]
type2 paramName3]...)]

; akcie

END ProcName
```

### Parametre

procname	in	Meno procedúry (musí vyhovovať pravidlám pre <a href="#">meno objektu</a> ).
type1, type2, ..., type10	in	Typ prvého (druhého, tretieho, ..., desiateho) formálneho parametra.
paramName1, paramName2, ..., paramName10	in	Meno prvého (druhého, tretieho, ..., desiateho) formálneho parametra.

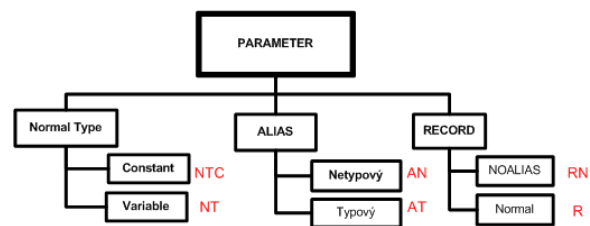
### Popis

Akcia uvádza procedúru (hlavica procedúry) s menom *ProcName*. Meno procedúry musí byť v rámci skriptu jednoznačné. Hlavica procedúry sa môže nachádzať pred [inicializovanou](#) [asou skriptu](#) mimo inej procedúry (vnorené procedúry nie sú podporené).

Procedúra môže obsahovať neobmedzený počet parametrov. Každý parameter v tele procedúry predstavuje lokálnu premennú (v deklarácii má uvedené meno aj typ). Ak je pred typom v deklarácii parametra uvedené kúové slovo **IN**, lokálna premenná je chápaná ako vstupná hodnota (prípadne zmeny jej hodnoty neovplyvnia hodnotu parametra po ukončení procedúry). Parametre bez kúového slova **IN** sú vstupno-výstupné.

Procedúra musí byť ukončená akciou **END ProcName**.

Možné typy formálnych parametrov procedúry:



Príklady - jednotlivé typy formálnych parametrov:

Typ formálneho parametra	Popis parametra	Príklad
NTC	Konštanta typu <i>Int</i> , <i>Bool</i> , ..., ako parameter nie je prípustná.	
NT	Premenná typu <i>Int</i> , <i>Bool</i> , <i>Real</i> , <i>Time</i> , <i>Text</i> .	INT _paramInt
AN	Netypový <i>ALIAS</i> .	ALIAS _an
AT	Typový <i>ALIAS</i> .	ALIAS (SD.RecordDef) _at
RN	Štruktúra bez odkazov na objekty.	RECORD NOALIAS (SD.RecordDef) _rn
R	Štruktúra.	RECORD (SD.RecordDef) _rn

Pri volaní procedúry akciou **CALL** sa na miesto formálneho parametra dosadí skutočný parameter. Pri formálnom parametri typu *NT*, prebehne typová konverzia medzi skutočným->formálnym->skutočným parametrom. Pre jednotlivé typy formálnych parametrov sú prípustné nasledovné kombinácie skutočných parametrov:

Kategória parametra	Formálny parameter					
	NT	AN	AT	RN	R	
IC_C	IN	-	-	-	-	
IC_HBJ_EXPR	IN	IN	-	-	-	
IC_L	-	-	-	-	-	
IC_L_CONST	IN	-	-	-	-	
IC_L_AN		-	-	-	-	
IC_L_AT_RIA		-	-	-	-	
IC_L_AT_RIN		-	-	-	-	
IC_L_AT_R	-	-	-	-	-	
IC_L_AT	-	IN	-	-	-	
IC_L_R_RIA		-	-	-	-	
IC_L_R_RIN		-	-	-	-	
IC_L_R_R	-	-	-	-	-	
IC_L_R	-	-	-	-	-	
IC_L_RNA_RIA		-	-	-	-	
IC_L_RNA_RIN		-	-	-	-	
IC_L_RNA_R	-	-	-	-	-	
IC_L_RNA	-	-	-	-	-	
IC_O		IN	-	-	-	
IC_O_R_RIA		-	-	-	-	
IC_O_R_RIN		-	-	-	-	
IC_O_R_R	-	-	-	-	-	
IC_O_R	-	-	-	-	-	
IC_L_NR_R	-	-	-	-	-	
IC_L_NR	-	-	-	-	-	
IC_L_NRNA_R	-	-	-	-	-	
IC_L_NRNA	-	-	-	-	-	

- Zelené políko označuje prípustnú kombináciu formálneho a skutočného parametra.
- Políko s textom IN označuje nutnosť deklarovať formálny parameter ako vstupný (kúové slovo **IN**).

innos so skutočným a formálnym parametrom pri volaní a návrate z procedúry:

Typ formálneho parametra	Skutoný parameter		Formálny parameter
	Volanie	Návrat*	Volanie
NT	ítanie hodnoty.	Nastavenie hodnoty.	Nastavenie hodnoty podľa parametra.
AN	Preítanie odkazu na objekt (nie hodnoty objektu).	Nastavenie odkazu na objekt (nie hodnoty).	Nastavenie odkazu na objekt podľa parametra, ktorý je tiež odkaz.
AT	Preítanie odkazu na objekt (nie hodnoty objektu).	Nastavenie odkazu na objekt (nie hodnoty).	Nastavenie odkazu na objekt podľa parametra, ktorý je tiež odkaz.
RN	ítanie hodnoty (riadok alebo celá štruktúra).	REDIM parametra podľa potreby (ak je index = 0 a skutoný parameter je lokálna premenná) a nastavenie hodnoty.	REDIM parametra podľa potreby a nastavenie hodnoty.
R	ítanie hodnoty (riadok alebo celá štruktúra).	REDIM parametra podľa potreby (ak je index = 0 a skutoný parameter je lokálna premenná) a nastavenie hodnoty.	REDIM parametra podľa potreby a nastavenie hodnoty.

\* len v prípade IN OUT parametra

Procedúru deklarovanú s kúovým slovom **RPC** je možné vola z iných ESL skriptov prostredníctvom [vzdial eného volania](#). Pre **RPC** procedúry sú prípustné typy formálnych parametrov **NT** a **RN**.

Vykonávanie [server eventu](#) (alebo skriptu) prebieha na jednej inštancii. Preto súasné volanie RPC procedúry viacerými inými udalosami alebo skriptami je serializované a požiadavky sa spracovávajú postupne prostredníctvom fronty požiadaviek.

Ak je vzdialená procedúra deklarovaná kúovým slovom **RPCX**, všetky požiadavky na vykonanie tejto procedúry sú vyradené z fronty pri príchode novej požiadavky. Výsledné správanie je potom také, že vo fronte neexistujú súasne dve požiadavky na vykonanie procedúry deklarovanej ako **RPCX**. Z pohadu volajúceho, vyradenie požiadavky z fronty znamená ukonenie vzdialeného volania (CALL) s chybou.

Kúové slovo **IMPLEMENTATION** je použité vtedy, ak procedúra implementuje uritú procedúru z objektu ESL Interface. Meno príslušného objektu je uvedené ako prefix mena procedúry, ktorý je oddelený znakom ^.

Kúové slovo **PUBLIC** je použité vtedy, ak je procedúra súasou eventu typu **UNIT**. Takýto typ procedúry je dovolené vola iba zo skriptu, v ktorom je daný UNIT definovaný (deklarovaný) ([volanie Public procedúr](#)).

Pomocou RPC procedúr je možný prenos [dátových kontajnerov](#) a [handle](#) na databázové spojenia.

## Formálny parameter procedúry NETYPOVÝ RECORD

Ak pri deklarácii formálneho parametra procedúry typu RECORD neuríme typ štruktúry (RECORD NOALIAS () \_rec), vznikne **netypový record**. Netypovému recordu urí typ skutoný parameter vstupujúci do procedúry pomocou akcie [CALL](#) pri volaní procedúry.

HBJ definície štruktúry netypového recordu je možné zisti pomocou ESL funkcie [%GetRecordStructHBJ](#) (IN recordVal) (\_hbj := % GetRecordStructHBJ(\_rec\HBJ)).

### Príklad

```
PROCEDURE Proc1(RECORD NOALIAS () _arr)
INT _iHbj

_iHbj := %GetRecordStructHBJ(_arr\HBJ)
; ....
; ....
; ....
REDIM _arr[ _arr\DIM + 1 ]
END

BEGIN
RECORD NOALIAS (SD.RecordDef) _arr1
; ....
; Nastavenie hodnoty lokálnej premennej _arr1 podľa hodnoty objektu
; SV.Struktura
REDIM _arr1[SV.Struktura\DIM]
SET _arr1 WITH SV.Struktura
CALL Proc1(_arr1)

END
```

Deklarácia procedúry a jej volanie s dvoma vstupnými a jedným výstupným parametrom:

```
PROCEDURE Scitaj(IN INT _p1, IN INT _p2, INT _v)
    _v := _p1 + _p2
    _p2:= _p2 + _p2
END Scitaj

BEGIN
    INT _a = 3
    INT _b = 2
    INT _c
    CALL Scitaj(_a, _b, _c)
END
```

Procedúra síta dva vstupné parametre *\_p1* a *\_p2* typu **INT**. Výsledok uloží do vstupno-výstupného parametra *\_v* typu **INT**. Po návrate z procedúry bude parameter *\_c* nastavený na súčet hodnôt parametrov *\_a* a *\_b*. Hodnota parametra *\_b* nebude zmenená akciou (*\_p2 := \_p2 + \_p2*), lebo *\_p2* je vstupný parameter a jeho zmeny sa nebudú zverejovať.

Procedúry je možné volať rekurzívne (procedúru z procedúry).

Deklarácia procedúry s parametrom typu štruktúra:

```
PROCEDURE Proc1(RECORD NOALIAS (SD.RecordDef) _arr)
; ....
; ....
; ....
    REDIM _arr[_arr+1]
END Proc1

BEGIN
    RECORD NOALIAS (SD.RecordDef) _arr1
; ....
; Nastavenie hodnoty lokálnej premennej _arr1 podľa hodnoty objektu
; SV.Struktura
    REDIM _arr1[SV.Struktura\DIM]
    SET _arr1 WITH SV.Struktura
    CALL Proc1(_arr1)

END
```

Volanie CALL v príklade prebieha v nasledovných krokoch:

1. Vyhodnotenie indexu riadku skutočného parametra (v tomto prípade 0 => celá hodnota objektu).
2. Preítanie hodnoty skutočného parametra *\_arr1*.
3. Zmena rozmeru formálneho parametra podľa potreby: REDIM *\_arr1*[*\_arr1*\DIM].
4. Naplnenie formálneho parametra: SET *\_arr* WITH *\_arr1*.

Návrat z procedúry *Proc1* podobne:

1. Preítanie hodnoty formálneho parametra *\_arr*.
2. Vyhodnotenie indexu riadku skutočného parametra (v tomto prípade 0 => celá hodnota objektu).
3. Zmena rozmeru formálneho parametra podľa potreby: REDIM *\_arr1*[*\_arr*\DIM].
4. Naplnenie formálneho parametra: SET *\_arr1* WITH *\_arr*.

Ak pri volaní nie je uvedený index pri skutočnom parametre (štruktúrovaná premenná alebo objekt typu Štruktúra), pracuje sa s celou hodnotou. Toto platí tiež vtedy, ak je uvedený index, ktorý nadobudne hodnotu = 0 (nesmie byť prístup na položku).

Výnimkou je skutočný parameter typu **IC\_L\_AT**. Tento je bez indexu a prístupu na položku chápaný ako odkaz na objekt typu [Štruktúrovaná premenná](#), nie ako jeho hodnota.

#### Poznámka

Normálne ukončenie procedúry prebehne vykonaním akcie **RETURN**, **END** MenoProcedury alebo predasne akciou **END** alebo vznikom chyby ak nie je v aktuálnej procedúre definovaná obsluha chyby (**ON ERROR**). Pri normálnom ukončení procedúry sú aktualizované vstupno-výstupné parametre v príslušnej akcii **CALL**.

Ak je procedúra volaná ako vzdialená a neprebehne u nej normálne ukončenie, nie je možné aktualizovať prípadné vstupno-výstupné parametre. Preto, ak je volaná synchronne s priradením, výsledkom priradenia je chyba *ERR\_MISSING\_RETURN*.



**Súvisiace stránky:**

[Akcie v skriptoch](#)