

# CALL - vzdialené volanie procedúr

## Akcia CALL - vzdialené volanie procedúr

Akcia **CALL** pre vzdialené volanie procedúr umožňuje vola procedúry, ktoré sú implementované v:

- skripte objektu typu [Event](#), ktorý musí byť typu [Server event](#),
- skripte aktívnej schémy (alebo podschémy),
- procese JAPI.

Volanie vzdialenej procedúry môže prebiehať nasledovne:

- **synchronne** - akcia **CALL** pokračuje až po ukončení vykonávania vzdialenej procedúry a modifikované vstupno-výstupné parametre budú aktualizované. Ak je akcia **CALL** zapísaná formou výrazu (`_i := CALL ...`) je možné získať prípadnú neošetrenú chybu (výnimku), ktorá vznikla v rámci volanej procedúry. Synchronne volania môžu vyvolať deadlock (uviaznutie). Dialóg s chybovým hlásením obsahuje [úplnú postupnosť volaní](#) (akcie CALL), ktoré viedli k deadlocku.
- **asynchronne** - akcia **CALL** generuje len požiadavku na vykonanie vzdialenej procedúry (neaká na jej ukončenie). Pri takomto volaní neexistuje možnosť kontroly vykonania procedúry.  
Asynchronne je možné vola:
  - RPC procedúru v konkrétnom skripte na konkrétnom procese,
  - RPC procedúru v skriptoch (objekty typu Event (označené ako "Server event") alebo Schéma), ktoré sú aktívne na všetkých bežiacich procesoch (**BROADCAST**) D2000 HI alebo D2000 Event Handler,
  - proces JAPI.

Deklarácia vzdialenej procedúry musí začínať kľúčovým slovom [RPC](#).

### Deklarácia - synchronne volanie

```
[_ret :=] CALL [objIdent] ProcName [(paramIdent1 [,paramIdent2]...)]  
[SYNC] [ON procIdent [INSTANCE instanceExpr]] [PRTY exprIntPrty]
```

### Deklarácia - asynchronne volanie

```
CALL [objIdent] ProcName [(paramIdent1 [,paramIdent2]...)] ASYNC [ON  
procIdent [INSTANCE instanceExpr]] [PRTY exprIntPrty]
```

### Deklarácia - asynchronne volanie BROADCAST

```
CALL [objIdent] ProcName [(paramIdent1 [,paramIdent2]...)] ASYNC ON ALL  
[PRTY prtyIdent]
```

### Deklarácia - synchronne volanie Klient a Server konfigurácia

```
[_ret :=] CALL [] ProcName [(paramIdent1 [,paramIdent2]...)] [SYNC] [ON  
procIdent [INSTANCE instanceExpr]] [PRTY exprIntPrty]
```

### Parametre

Procname	in	Meno procedúry (musí vyhovovať pravidlám pre <a href="#">meno objektu</a> ).
paramIdent1, paramIdent2, ..., paramIdentN	in	<a href="#">Identifikátor</a> hodnoty pre prvý (druhý, tretí, ..., N-tý) parameter. Počet parametrov musí byť zhodný s počtom parametrov volanej procedúry.

objIdent	in	<ul style="list-style-type: none"> <li>Identifikátor objektu (schémy alebo systémového server eventu), ktorý obsahuje skript s danou procedúrou.</li> <li>Lokálna premenná, ktorej hodnota je zhodná s hodnotou premennej typu <i>RefId</i>, ktorá je pripojená ku grafickému objektu typu <i>Schéma</i> (volanie procedúry podschémy).</li> <li>Pri volaní RPC z Lokálneho skriptu do Remote skriptu (<b>HIS Server</b>) sa identifikátor <i>objIdent</i> vynecháva.</li> <li>Pri volaní procesu JAPI, musí by odkaz na objekt prázdny (<b>pozri príklad</b>).</li> </ul>
procIdent	in	Identifikátor procesu, kde sa objekt <i>objIdent</i> nachádza (napr. SELF.EVH, SELF.HIP alebo WS_BC.HIP). Pokiaľ je akcia CALL použitá v rámci schémy, tak <b>preddefinovaná premenná</b> <i>_FROM_HIP</i> udáva identifikátor procesu HI, v ktorom je schéma otvorená.
instanceExpr	in	Nepovinný výraz typu <i>Int</i> . Adresuje pri volaní procedúry určitú inštanciu.
prtyIdent	in	Identifikátor typu <i>Int</i> . Uruje prioritu vykonania RPC procedúry. Parameter môže nadobúdať hodnotu z intervalu: <IntegerRange> (-2147483648 to 2147483647). Ak parameter nie je zadáný, jeho preddefinovaná hodnota je 0. Prednostne spracovaná procedúra je procedúra s vyššou prioritou (s vyšším islom).

## Popis

Akcia **CALL** vykoná volanie **RPC** (RPCX) procedúry s menom *ProcName*. Za menom procedúry nasleduje v zátvorkách zoznam parametrov oddelených iarkami. Poet (a typy) parametrov musia byť zhodné s potom parametrov volanej procedúry (ak nie je zhodný počet parametrov je generovaná výnimka *\_ERR\_INV\_NUM\_PARAMS*). Ak je niektorý parameter v deklarácii procedúry označený ako vstupno-výstupný, zodpovedajúci parameter pri volaní procedúry nesmie byť konštanta (v prípade chyby je generovaná výnimka *\_ERR\_SET\_CONST*). Parameter *procIdent* je odkaz na objekt typu **Proces**, na ktorom je objekt *objIdent* otvorený. Pre objekt typu **Event** je to proces **D2000 EventHandler**, ktorý je jeho **rodiom** (parent) alebo proces, na ktorom je volaný Event otvorený (akcia **OPENEVENT**). Pre objekt typu **D2000 HI**, v ktorom je schéma otvorená. Ak zadáný proces nemá otvorený žiadaný objekt, event generuje chybu *\_ERR\_OBJECT\_NOT\_FOUND*.

Parameter **INSTANCE** uruje číslo inštancie objektu (schémy alebo eventu).

Ak je pre identifikátor *objIdent* použitá lokálna premenná typu *RefId*, parametre **INSTANCE** a **ON** nie sú prípustné (ich hodnoty sú dané kontextom volania).

Ak sa volá RPC procedúra medzi Lokálnou a Remote asou "Klient a server" konfigurácie eventu (**HIS Server**), identifikátor *objIdent* sa vynecháva.

```
CALL [ ] ProcName [(paramIdent1 [,paramIdent2]...)] ...
```

Chyby označené znakom \* sú generované na strane volaného skriptu. Volajúci skript (akcia **CALL**) túto chybu môže zistiť jedine pri synchrónnom volaní a pri zápise akcie s priradením. Ak nastane chyba, jej kód bude naplnený do premennej *\_ret* ktorá musí byť typu **INT**.

Možné typy parametrov: pozri akciu **PROCEDURE**.

Volanie typu **BROADCAST** je povinné asynchrónne. Ako meno procesu (parameter *procIdent*) sa uvedie kúové slovo **ALL**.

Pri vykonávaní takéhoto volania systém automaticky distribuuje požiadavku na vykonanie procedúry na všetky bežiacie procesy **D2000 HI** alebo **D2000 EventHandler**. Tieto procesy vyhľadajú všetky inštancie (alebo základné objekty) skriptov, ktoré sú identifikované parametrom *objIdent* a generujú pre ne požiadavky na vykonanie procedúry *ProcName* s príslušnými parametrami.

Kúové slovo **INSTANCE** nie je pri tomto type volania povolené.

Od verzie V8.00.008 R9 má akcia **CALL** implementovanú optimalizáciu prenosu formálnych parametrov. Táto optimalizácia zabezpečí, že formálny parameter RPC procedúry bude "odkaz" na skutočný parameter. Priamy dôsledok je podstatné zvýšenie rýchlosti volania RPC procedúr.

Podmienky:

- volanie prebieha na rovnakom procese \*.EVH,
- volanie procedúry je synchrónne,
- parameter musí byť typu *Record*
- parameter musí byť *IN/OUT*.

Kúové slovo **PRTY** umožňuje zadanie priority vykonania RPC procedúry. Táto vlastnosť zabezpečí prioritné vykonanie RPC procedúry. Priorita je daná parametrom *prtyIdent* za kúovým slovom **PRTY**.

Pomocou RPC procedúr je možný prenos **dátových kontajnerov** a handle na **databázové spojenia**.

Pri volaní procesu JAPI platia nasledujúce pravidlá:

- parameter *objIdent* je prázdny objekt ([pozri príklad](#)),
- nie je možné použiť kúové slovo INSTANCE,
- nie je možné vola RPC procedúru, ktorá je implementáciou ESL Interface,
- identifikátor procesu (*proclIdent*) musí byť typu IC\_HOBJ\_EXPR (výraz typu [HOBJ](#)),
- príjem volaní na strane procesu JAPI musí byť implementovaný ako listener registrovaný metódou D2Session::setRPCListener.

## Príklad

Synchronné volanie vzdialenej procedúry systémového eventu:

```
INT _i
TEXT _personName
_personName := "Peter"
_i := CALL [E.Work] AddPerson(_personName) SYNC ON SELF.EVH
IF _i # _ERR_NO_ERROR THEN
    ; chyba pri volaní vzdialenej procedúry
ENDIF
```

Asynchronné volanie vzdialenej procedúry aktívnej schémy:

```
TEXT _msg
_msg := " ... "
CALL [S.Picture] SendMessage(_msg) ASYNC ON _FROM_HIP
```

Asynchronné volanie vzdialenej procedúry aktívnej schémy:

```
TEXT _msg
_msg := " ... "
CALL [_podSchema] SendMessage(_msg) ASYNC
```

Synchronné volanie vzdialenej procedúry, ktorú implementuje:

```
TEXT _msg
INT _hbj
_msg := " ... "
_hbj := ..... hodnota získaná napríklad pri volaní RPC procedúry z JAPI
klienta, kde jeden z parametrov je HOBJ príslušného procesu JAPI.
CALL [(0)] SendMessage(_msg) SYNC ON (_hbj)
```

Asynchronné volanie BROADCAST:

```
CALL [E.BROADCAST_RECEIVER] Broadcast(1) ASYNC ON ALL
```

Synchronné volanie vzdialenej procedúry so zadanou prioritou:

```
INT _i
INT _prty
TEXT _personName

_personName := "Peter"
_prty := 100
_i := CALL [E.Work] AddPerson(_personName) SYNC ON SELF.EVH PRTY _prty
IF _i # _ERR_NO_ERROR THEN
    ; chyba pri volaní vzdialenej procedúry
ENDIF
```

Synchronné volanie vzdialenej procedúry z Lokálneho do Remote skriptu:

```
INT _i
TEXT _personName
_personName := "Peter"
_i := CALL [] AddPerson(_personName) SYNC ON SELF.HIP
IF _i # _ERR_NO_ERROR THEN
; chyba pri volaní vzdialenej procedúry
ENDIF
```

#### Poznámka

Volaná vzdialená procedúra musí byť ukonená akciou **RETURN**, alebo **END MenoProcedury**. V opanom prípade nie je možné aktualizovať prípadné vstupno-výstupné parametre. Pri synchronnom volaní s priradením je návratová hodnota nastavená na chybu **\_ERR\_MISSING\_RETURN**.



#### Súvisiace stránky:

[Akcie v skriptoch](#)

[Akcia CALL - lokálne volanie procedúr](#)

[Prenos handle na databázové spojenie medzi bežiacimi ESL skriptami](#)

[Prenos dátových kontajnerov medzi bežiacimi ESL skriptami](#)