CALL - vzdialené volanie procedúr

Akcia CALL - vzdialené volanie procedúr

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

- skripte objektu typu Event, ktorý musí by typu Server event,
- skripte aktívnej schémy (alebo podschémy),
- procese JAPI.

Volanie vzdialenej procedúry môže prebieha nasledovne:

- **synchrónne** - akcia **CALL** poká na ukonenie vykonávania vzdialenej procedúry a modifikované vstupno-výstupné parametre budú aktualizované. Ak je akcia **CALL** zapísaná formou výrazu (_i := CALL ...) je možné zisova prípadnú neošetrenú chybu (výnimku), ktorá vznikla v rámci volanej procedúry. Synchrónne volania môžu vyvola deadlock (uviaznutie). Dialóg s chybovým hlásením obsahuje úplnú postupnos volaní (akcie CALL), ktoré viedli k deadlocku.
- **asynchrónne** - akcia **CALL** generuje len požiadavku na vykonanie vzdialenej procedúry (neaká na jej ukonenie). Pri takomto volaní neexistuje možnos kontroly vykonania procedúry.
  Asynchrónne je možné vola:
  - RPC procedúru v konkrétnom skripte na konkrétnom procese,
  - RPC procedúru v skriptoch (objekty typu Event (oznaené 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 zaína kúovým slovom RPC.

Deklarácia - synchrónne volanie

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

Deklarácia - asynchrónne volanie

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

Deklarácia - asynchrónne volanie BROADCAST

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

Deklarácia - synchrónne volanie Klient a Server konfigurácia

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

Parametre

Procname	in	Meno procedúry (musí vyhovova pravidlám pre meno objektu).
paramIdent1, paramIdent2, ..., paramIdentN	in	Identifikátor hodnoty pre prvý (druhý, tretí, ..., N-tý) parameter. Poet parametrov musí by zhodný s potom parametrov volanej procedúry.

objIdent	in	• Identifikátor objektu (schémy alebo systémového server eventu), ktorý obsahuje skript s danou procedúrou. • Lokálna premenná, ktorej hodnota je zhodná s hodnotou premennej typu *RefId*, ktorá je pripojená ku grafickému objektu typu *Schéma* (volanie procedúry podschémy). • Pri volaní RPC z Lokálneho skriptu do Remote skriptu (HIS Server) sa identifikátor *objIdent* vynecháva. • Pri volaní procesu JAPI, musí by odkaz na objekt prázdny (pozri príklad).
procIdent	in	Identifikátor procesu, kde sa objekt *objIdent* nachádza (napr. SELF.EVH, SELF.HIP alebo WS_BC.HIP). Pokia je akcia CALL použitá v rámci schémy, tak preddefinovaná premenná _FROM_HIP udáva identifikátor procesu HI, v ktorom je schéma otvorená.
instanceExpr	in	Nepovinný výraz typu *Int*. Adresuje pri volaní procedúry uritú inštanciu.
prtyIdent	in	Identifikátor typu *Int*. Uruje prioritu vykonania RPC procedúry. Parameter môže nadobúda hodnotu z intervalu: <Integer'Range> (-2147483648 to 2147483647). Ak parameter nie je zadaný, jeho preddefinovaná hodnota je 0. Prednostne spracovaná procedúra je procedúra s vyššou prioritou (s vyšším íslom).

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 by zhodné s potom parametrov volanej procedúry (ak nie je zhodný poet parametrov je generovaná výnimka *_ERR_INV_NUM_PARAMS**).
Ak je niektorý parameter v deklarácii procedúry oznaený ak vstupno-výstupný, zodpovedajúci parameter pri volaní procedúry nesmie by 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 zadaný proces nemá otvorený žiadaný objekt, event generuje chybu *_ERR_OBJECT_NOT_FOUND**.

Parameter **INSTANCE** uruje íslo 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 oznaené znakom * sú generované na strane volaného skriptu. Volajúci skript (akcia CALL) túto chybu môže zisti jedine pri synchrónnom volaní a pri zápise akcie s priradením. Ak nastane chyba, jej kód bude naplnený do premennej *_ret* ktorá musí by typu **INT**.

Možné typy parametrov: pozri akciu PROCEDURE.

Volanie typu **BROADCAST** je povinne asynchrónne. Ako meno procesu (parameter *procIdent*) sa uvedie kúové slovo **ALL**.
Pri vykonávaní takéhoto volania systém automaticky distribuuje požiadavku na vykonanie procedúry na všetky bežiace procesy D2000 HI alebo D2000 EventHandler. Tieto procesy vyhadajú 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 zabezpeí, že formálny parameter RPC procedúry bude "odkaz" na skutený parameter. Priamy dôsledok je podstatné zvýšenie rýchlosti volania RPC procedúr.
Podmienky:

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

Kúové slovo **PRTY** umožuje zadanie priority vykonania RPC procedúry. Táto vlastnos zabezpeí prioritné vykonanie RPC procedúry. Priorita je zadaná 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 použi kúové slovo INSTANCE,
- nie je možné vola RPC procedúru, ktorá je implementáciou ESL Interface,
- identifikátor procesu (*procIdent*) musí by typu IC_HOBJ_EXPR (výraz typu HOBJ),
- príjem volaní na strane procesu JAPI musí by implementovaný ako listener registrovaný metódou D2Session::setRPCListener.

Príklad

Synchrónne 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
```

Asynchrónne volanie vzdialenej procedúry aktívnej schémy:

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

Asynchrónne volanie vzdialenej procedúry aktívnej schémy:

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

Synchrónne 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)
```

Asynchrónne volanie BROADCAST:

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

Synchrónne 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
```

Synchrónne 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 by ukonená akciou RETURN, alebo **END MenoProcedury**. V opanom prípade nie je možné aktualizova prípadné vstupno-výstupné parametre. Pri synchrónnom 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