Akcie v skripte

Akcie v skriptoch

Skript je tvorený postupnosou akcií, ktoré sa po aktivácii vykonajú. Akcie sa zapisujú v prostredí editora akcií.
asti zápisu uvedené v hranatých zátvorkách **[ ]** sú nepovinné. Znak **|** (t.j. ALEBO) vyjadruje možnos alternatívneho zápisu.

Typy akcií

Zoznam jednotlivých typov akcií. Nasledujúci zoznam popisuje základnú množinu akcií, ktoré sú prístupné v obidvoch aplikáciách skriptu (objekt typu Event , Aktívna schéma).
Akcie možno rozdeli do týchto základných kategórií (typov):

- priraovacie akcie
- akcie pre prístup k databáze
- akcie pre obsluhu chybových stavov
- riadiace akcie
- akcie pre komunikáciu s operátorom
- akcie pre prácu s archívom
- akcie pre synchronizovanie vykonávania akcií skriptov
- riadenie alarmov
- akcie pre prácu so štruktúrami
- akcie pre prácu s dátovými kontajnermi
- akcie pre riadenie prístupových práv
- akcie pre prácu so zoznamami objektov
- ostatné akcie

Poznámka: V systéme D2000 je možné implementova aplikanú funkcionalitu použitím jazyka JAVA. Ekvivalenty k ESL akciám sú uvedené v samostatnej on-line dokumentácii, ktorá sa nachádza v podadresári **Help** programového adresára systému D2000.

Priraovacie akcie

Priraovacie akcie umožujú meni hodnoty a odkazy na objekty:

1. objektov v systéme
2. lokálnych premenných

- Priradenie
- SET WITH
- SET AS [DIRECT]
- SET BIND

Akcie pre prístup k databáze

Pre prácu s tabukou databázy sú urené skupiny akcií, ktoré používajú rôzne spôsoby prístupu k dátam.
Ak pri práci s databázou nastane chyba, íselný kód ktorý ju bližšie popisuje je možné získa volaním funkcie %GetLastExtErrorCode. Podrobnejšie informácie o chybe je možné získa funkciou %GetLastExtErrorMsg.
Prvý spôsob využíva existenciu kúa alebo podmienku WHERE. Poda spôsobu zápisu akcie umožuje zápis/ítanie jedného alebo viacerých riadkov.

- DB_DELETE
- DB_CONNECT
- DB_DISCONNECT
- DB_INSERT
- DB_INSUPD
- DB_READ
- DB_READ_BLOB
- DB_SET_PROCESS_PARAMS
- DB_UPDATE
- DB_UPDATE_BLOB

Príklad práce s tabukou (akcie DB_...)

Druhý spôsob pracuje pomocou stránkovania, priom vekos stránky (poet riadkov v tabuke) je volitený.

- PG_CONNECT
- PG_DISCONNECT
- PG_READ
- PG_INSERT
- PG_DELETE
- PG_UPDATE

Príklad práce s tabukou (akcie PG_ ...)

Tretí spôsob je modifikácia prvého. Modifikácia spoíva v tom, že práca s tabukou nevyžaduje akcie typu CONNECT a DISCONNECT.

- DBS_DELETE
- DBS_INSERT
- DBS_INSUPD
- DBS_READ
- DBS_READ_BLOB
- DBS_UPDATE
- DBS_UPDATE_BLOB

Posledný spôsob umožuje využi celú škálu **SQL** príkazov pri práci s databázou.

- SQL_CONNECT
- SQL_DISCONNECT
- SQL_EXEC_DIRECT
- SQL_EXEC_PROC
- SQL_SELECT

ítanie databázy prostredníctvom príkazu **SELECT**.

- SQL_BINDIN
- SQL_FETCH
- SQL_FREE
- SQL_PREPARE

Príklad práce s databázou (akcie SQL_ ...)

Pre riadenie transakného spracovania príkazov (v zmysle databázovej transaknosti) je možné použi nasledovné akcie:

- DB_TRANS_OPEN
- DB_TRANS_COMMIT
- DB_TRANS_ROLLBACK
- DB_TRANS_CLOSE

Príklad transaknej práce s databázou

Vynútenie obnovy (refresh) zobrazených dát v pohadoch užívatea v procese D2000 HI (napríklad zobrazova typu Browser) je možné vyvola akciou:

- DB_REFRESH_TABLE

Zaregistrovanie procedúry, ktorá sa zavolá po

- zmene dát v tabuke,
- zrušení tabuky,
- pri prepínaní aktívnej a pasívnej inštancie procesu DbManager,
- obnove dát akciou DB_REFRESH_TABLE,

je možné akciou:

- ON DB_CHANGE

Prenos handle na databázové spojenie medzi bežiacimi ESL skriptami

Konverzia a reprezentácia hodnôt v databáze

Zápis

Hodnota premennej (vo väšine prípadov) políka lokálnej štruktúrovanej premennej je pri zápise do databázy konvertovaná tak, že ak je **platná**, je zapísaná normálnym spôsobom. Ak je **neplatná** je do databázy vložená NULL hodnota. Pri type **TEXT** toto pravidlo funguje rovnako, okrem databázy ORACLE, kde je prázdny text reprezentovaný databázovou hodnotou NULL rovnako ako neplatná hodnota.

ítanie

Po preítaní hodnoty z databázy (ktorá je rôzna od NULL) prebehne konverzia hodnoty na požadovaný typ, ktorý je daný typom premennej, do ktorej je umiestnený výsledok ítania. V prípade úspešnej konverzie je výsledná hodnota platná. Pri ítaní NULL hodnoty je výsledná hodnota neplatná. Pri type **TEXT** je NULL hodnota v databáze konvertovaná na platný prázdny textový reazec. Jediná výnimka je ítanie prostredníctvom akcie DB_READ/DBS_READ na platforme ORACLE OCI, kedy je NULL hodnota konvertovaná na **neplatnú** hodnotu.

Tabuka znázoruje výsledok zápisu a ítania textovej hodnoty v závislosti na databázovej platformy.

DBS_INSERT - zápis textovej hodnoty do databázy (D2Value -> DBValue).
PG_READ, BrowserRead - ítanie textovej hodnoty z databázy akciou PG_READ alebo do zobrazovaa Browser (dáta zverejnené cez OnFetchDone) (DBValue -> D2Value).
DB_READ - ítanie textovej hodnoty prostredníctvom akcie DB_READ (DBValue -> D2Value).

Databáza	DBS_INSERT	PG_READ, BrowserRead	DB_READ
Sybase 12/PostgreSQL	"Text" -> "Text	"Text" -> "Text	"Text" -> "Text
dbmanager.exe	"" -> ""	"" -> ""	"" -> ""
	Invalid->NULL	NULL->""	NULL->""
ORACLE OCI	"Text" -> "Text	"Text" ->"Text	"Text" -> "Text
dbmanager_ora.exe	"" -> NULL		
	Invalid->NULL	NULL->""	NULL->Invalid
ORACLE ODBC	"Text" -> "Text	"Text" -> "Text	"Text" -> "Text
dbmanager.exe	"" -> NULL		
	Invalid->NULL	NULL->""	NULL->""

Akcie pre obsluhu chybových stavov

- EXCEPTION_HANDLER
- ON ERROR
- RESUME
- RETRY

Riadiace akcie

Riadiace akcie sú akcie, ktoré ovplyvujú tok riadenia (poradie vykonávania akcií) v skripte.

- BEGIN
- CALL - lokálne volanie procedúr
- CALL - vzdialené volanie procedúr
- CALL - volanie Public procedúr
- DELAY
- DO_LOOP, EXIT_LOOP, END_LOOP
- ENABLE
- END
- END procedure
- EVENT
- GOSUB
- GOTO
- IF GOTO
- IF THEN [ELSE] ENDIF
- IMPLEMENTATION
- ON DB_CHANGE
- ON GOTO
- ON CHANGE
- OnExternalEvent
- PRAGMA
- PROCEDURE
- RETURN
- Riadiace funkcie (bez návratovej hodnoty)
- WAIT

Akcie pre komunikáciu s operátorom

Nasledovné akcie umožujú implementova v skripte dialóg s operátorom, alebo ovláda schémy na pracovnej konzole operátora. Tu je vhodné použi preddefinovanú lokálnu premennú *_FROM_HIP*. Ak je skript spustený zo schémy (pripojený grafický objekt na ovládanie), je automaticky táto lokálna premenná asociovaná s procesom, z ktorého bol skript odštartovaný. Toto umožuje adresne pracova s týmto procesom:

- otvára, zatvára schémy,
- posiela správy pre operátora,
- smerova akciu QUERY.

- CLOSE
- MESSAGE
- OPEN
- OPENEVENT
- QUERY

Akcie pre synchronizovanie vykonávania akcií skriptov

Akcie GETACCESS a RELEASEACCESS umožujú navzájom synchronizova vykonávanie akcií v:

1. rôznych inštanciách eventov v rámci jedného procesu D2000 Event Handler,
2. rôznych skriptoch aktívnych schém v rámci jedného procesu D2000 HI,
3. rôznych skriptoch alebo globálne v aplikácii v inštanciách eventov alebo skriptov aktívnych schém.

Poskytujú uritú formu komunikácie medzi skriptami.

- GETACCESS
- RELEASEACCESS

Akcie pre prácu s archívom

ítanie/zapisovanie dát z/do archívu, mazanie dát v archíve.

- CALCARCHEXPR
- CALCONDEMANDSTAT
- CALCSTATFUNC
- CALCSTATFUNCARR
- DELETEARCHDATA
- GETARCHARR
- GETARCHARR_TO_CNT
- GETARCHCOL
- GETARCHROW
- GETARCHSTRUCT
- GETARCHVAL
- INSERTARCHARR
- UPDATEARCHVAL

Ovládanie alarmov

Ovládanie systémových alebo procesných alarmov.

- BLOCK
- KVIT
- UNBLOCK
- UNBLOCK_ALL

Akcie pre prácu so štruktúrami

Pri práci s rozsiahlymi lokálnymi štruktúrami je obas potrebné štruktúru utriedi, vloži alebo zmaza riadok, alebo vyhada riadok. ESL toto umožuje, ale tieto operácie si vyžadujú prechod celej štruktúry cyklom. Toto môže by asovo nároné. ESL preto definuje nasledujúce akcie, ktoré uvedené innosti vykonávajú optimálnejšie:

- COPYCOL
- COPYCOLIDX
- DELETE
- EXPORT_CSV
- EXPORT_CSV_TEXT
- FIND_TRUE
- GETCOLTIME
- GETROWDESC
- IMPORT_CSV
- INSERT
- SETCOLTIME
- SORT
- TRANSCOLTOROW
- TRANSROWTOCOL

Akcie pre prácu s dátovými kontajnermi

Akcie umožujú prácu so skladom hodnôt, tzv. *dátovým kontajnerom* (interná dátová štruktúra).

- CNT_CNVTOARRAY
- CNT_CREATE
- CNT_DEBUG
- CNT_DELETE
- CNT_DESTROY
- CNT_FIND
- CNT_GETITEM
- CNT_GETKEY

- CNT_GETNR
- CNT_INSERT

Akcie **CNT_GETNR**, **CNT_CNVTOARRAY** a **CNT_GETITEM** umožujú prezeranie hodnôt z kontajnera pomocou indexu. Akcia **CNT_CNVTOARRAY** interne vytvorí pole, v ktorom sú všetky hodnoty usporiadané poda kúa vzostupne. Index je poradové íslo hodnoty v rámci poa.
Pole zaniká zmazaním alebo vložením hodnoty do kontajnera.

Vznik kontajnera zabezpeujú akcie **CNT_CREATE** - vone použitený kontajner, alebo **GETARCHARR_TO_CNT**. Druhý typ kontajnera je plnený stránkami, ktoré obsahujú dáta preítané z archívu.
Takto koncipovaný prístup do archívu je efektívnejší (spotreba pamäte a iastone rýchlos) ako použitie akcie GETARCHARR. Pre kontajner vytvorený akciou GETARCHARR_TO_CNT sú použitené len akcie CNT_GETNR, CNT_FIND a CNT_DESTROY (príklad).

Prenos dátových kontajnerov medzi bežiacimi ESL skriptami

Akcie pre riadenie prístupových práv

Akcie umožujú nastavenie prístupových práv poas behu systému.

- REBUILD_ACC
- RES_GROUP_DELETE
- RES_GROUP_DELETE_ALL
- RES_GROUP_INSERT
- RES_GROUP_QUERY

Akcie pre prácu so zoznamami objektov

Akcie pre prácu so zoznamami objektov umožujú:

- vytvori zoznam objektov poda zadaných kritérií,
- prejs na prvú, predošlú, nasledujúcu alebo poslednú stránku zoznamu,
- prejs na stránku urenú jej poradovým íslom,
- zisti celkový poet objektov zoznamu (pozor! nejde o poet objektov na stránke),
- zatvori zoznam.

Po vytvorení zoznamu sú ihne sprístupnené dáta na prvej stránke.
Taktiež prechod na prvú, predošlú, nasledujúcu, poslednú alebo poradovým íslom zadanú stránku sprístupuje dáta danej stránky.
Každý záznam v zozname objektov predstavuje jednoznaný identifikátor objektu, názov objektu, popis objektu, typ objektu, poet riadkov a poet stpcov.

- LST_CREATE
- LST_CLOSE
- LST_GO_NEXT
- LST_GO_PREV
- LST_GO_LAST
- LST_GO_FIRST
- LST_GO_PAGE
- LST_GETINFO

Príklad práce so zoznamami objektov (akcie LST_...).

Ostatné akcie

- COMMAND
- COPYOBJECT
- DELETEOBJECT
- FIND_FILES
- GETOLDVAL
- GETPOINTADR
- LOG
- LOGEX
- PLAY
- READLOG
- REDIM
- RUN
- RUNEX
- SETDT_LINEOBJ