

Structura scripturilor de Salvare/Restaurare

Scripturile de salvare/restaurare au o structură standardizată, organizată pe secțiuni, fiecare secțiune reprezentând un script apelabil separat. Deoarece operația în sine are mai multe etape, scriptul fiind apelat de mai multe ori, de fiecare dată se execută segmentul de scripturi dedicat evenimentului în curs de execuție.

Înainte de apel se încarcă și o structură cu parametrii actuali, care în script poate fi citită, interpretată, operațiile putând fi ramificate în funcție de conținutul actual al parametrilor.

Parametri de intrare/ieșire

Pentru comunicarea între secțiunile din cadrul pachetului sunt accesibile variabilele globale X_1, X_2, ..., X_9. Valorile încărcate în variabilele globale se vor păstra pe durata unei sesiuni de lucru. La părăsirea utilitarului, valorile se pierd.

Înainte de fiecare apel se creează h_GCmParam cu următoarele elemente:

```
h_GCmParam[ 1] - cod pachet: COPx
h_GCmParam[ 2] - rezultatul operației.
                  Parametru de ieșire - se încarcă în script.
                  .T. operație reușită; .F. - operație nereușită.
                  Dacă operația de montare nu este reușită, nu se
                  mai continuă.
h_GCmParam[ 3] - calea de acces către catalog arhiva în care
                  se vor crea pachetele arhivă sau
                  în care sunt pachetele de restaurat (ex. E:\DEVROL),
h_GCmParam[ 4] - nume fișier arhivă (ex. A:\DEVROL\CNTIAN04.ZIP),
h_GCmParam[ 5] - calea de acces către baza de date care se salvează
                  sau în care se restaurează pachetul (ex. C:\HAMOR\DBF),
h_GCmParam[ 6] - machetă pachete. Necesară pentru identificare pachete
                  (*.ZIP - valoare implicită,
                  alte ex. care se încarcă explicit în script: *.ARJ)
h_GCmParam[ 7] - la salvare: nume fișier script (lista generică a fișierelor
                  ce vor fi împachetate),
```

Structura scriptului de salvare/restaurare

Scriptul de salvare/restaurare este segmentat, structura generală a scriptului fiind cea de mai jos:

#MONTARE

```
// se execută pregătirea suportului pe care se va executa salvarea:
// -alegerea destinației unde se va face salvarea,
// -montarea dispozitivului (dacă este cazul),
// -verificări, dacă destinația aleasă este sau nu disponibilă,
// -localizarea utilitarelor de salvare propriu-zisă (programele de
// împachetare/despachetare pkzip, pkunzip, arj, etc.)
```

#SFIRSIT_MONTARE

#SALVARE

```
// Se execută operația de salvare propriu-zisă, care poate diferi în
// funcție de:
// -utilitarul de împachetare ales,
// -de sistemul de operare sub care se execută salvarea,
// Tot în acest segment poate fi implementată și afișarea unui fișier log
// cu privire la desfășurarea operației de salvare.
```

#SFIRSIT_SALVARE

```

#RESTAURARE
// Se execută operația de restaurare propriu-zisă, care poate diferi în
// funcție de:
// -utilitarul de restaurare ales,
// -de sistemul de operare sub care se execută restaurarea,
// Tot în acest segment poate fi implementată și afișarea unui fișier log
// cu privire la desfășurarea operației de restaurare.
#SFIRSIT_RESTAURARE

#VIZUALIZARE
// Se apelează pentru vizualizarea conținutului unui pachet.
// Numele pachetului se citește din lista de parametri transmiși, conținutul
// citit se afișează într-o fereastră text pe ecran.
// Citirea conținutului se face cu ajutorul utilitarului de
// salvare/restaurare ales.
#SFIRSIT_VIZUALIZARE

#STERGERE
// Se implementează ștergerea/formatarea dispozitivului curent.
// Bineînțeles, este în funcție de dispozitivul ales (CD, flash, floppy, etc.)
// și de sistemul de operare sub care se rulează utilitarul.
#SFIRSIT_STERGERE

#DEMONTARE
// Operații de terminare, dacă este cazul, ex. demontarea dispozitivului.
#SFIRSIT_DEMONTARE

```

Segmentele de scripturi se execută în următoarea ordine:

- Montare (dacă nereușită, nu se mai continuă),
- Salvare, Restaurare, Vizualizare, Ștergere- unul dintre segmentele enumerate, în conformitate cu operația curentă,
- Demontare,

Segmentele dedicate

Prezentăm un exemplu, dedicat pentru testarea și înțelegerea funcționării scriptului de salvare/restaurare.

Scriptul este disponibil în bazele de date la livrare. Este scriptul *COP2-Salvare/Restaurare-parametrii* fiind închis la livrare. Pentru testare deschideți scriptul cu hMET2C, după care va fi accesibil între operațiile de salvare/restaurare și poate fi testat.

```

#MONTARE
h_GCmdParam[ 6] := "*.ZIP"
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])
AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,"MONTARE","HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_MONTARE

#SALVARE
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])

```

```

AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,,"SALVARE","HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_SALVARE

#RESTAURARE
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])
AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,,"RESTAURARE",NIL,"HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_RESTAURARE

#VIZUALIZARE
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])
AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,,"VIZUALIZARE","HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_VIZUALIZARE

#STERGERE
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])
AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,,"STERGERE","HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_STERGERE

#DEMONTARE
aMes := {}
AADD (aMes, "1:"+h_GCmdParam[1])
AADD (aMes, "2:"+IF(h_GCmdParam[2]=.T., "T", "F"))
AADD (aMes, "3:"+h_GCmdParam[3])
AADD (aMes, "4:"+h_GCmdParam[4])
AADD (aMes, "5:"+h_GCmdParam[5])
AADD (aMes, "6:"+h_GCmdParam[6])
AADD (aMes, "7:"+h_GCmdParam[7])
aMesHlp := h_NewMessage("",,,,,,"DEMONTARE","HELP", aMes)
h_PopUpWindow (aMesHlp)
#SFIRSIT_DEMONTARE

// -----

```