

Descriere funcției xls/xml

1. FUNCȚIA h_ExpXlsXml

Exportă un set de date în format XML Spreadsheet

Parametri:

- pc_TipOp
 - Obligatoriu
 - Tip: Character 1
 - Valori Posibile: ""|" "<"|"-|">"
 - daca "", sau " " - se crează fișierul XML pentru toate înregistrările din zona curentă
 - daca "<" - se inițializează crearea
 - daca "- " - se exportă numai înregistrarea curentă
 - daca "<" - se închide crearea
- pa_Antet
 - Obligatoriu
 - Tip: Array
 - Forma: {[rand1 [,....,randn]]}
 - În cazul pc_TipOp = ">"|"- " poate fi și NIL
 - Poate să fie și array vid {} - în acest caz nu se exportă nimic
 - Rand
 - Optional
 - Tip:Array
 - Forma: {[text],[alinie]}
 - Text
 - Optional
 - Tip: Character
 - Valoare Implicită: ""
 - Alinie
 - Optional
 - Tip: Character 1
 - Valori Posibile: "L"|"R"|"C"
 - Valoare Implicită:"L"
 - Fiecare element reprezintă un rând de Antet

- pa_Camp
 - Obligatoriu
 - Tip:Array
 - Forma: {Coloana1[,...,[Coloana1]]}
 - În cazul pc_TipOp = ">"|"-" poate fi și NIL
 - Coloana
 - Tip: Array
 - Forma : {aCapTabel,cColTip,cCol,[cColDType],[nColLenght],[nColDec],[cColPict],[lColWrap]} - in cazul cColTip
- = "F"
 - {aCapTabel,cColTip,cCol, cColDType, nColLenght, nColDec, [cColPict],[lColWrap]} - in cazul cColTip = "E"
 - CapTabel
 - Obligatoriu
 - Tip: Array
 - Forma:{"Rand1"[,..."Randn"]}]}
 - Rand - Tip: Caracter
 - Reprezintă un rând din capul de tabel
 - Reprezintă rândurile capului de tabel
 - cColTip
 - Obligatoriu
 - Tip: Caracter 1
 - Valori Posibile "F" - Field|"E" - Expresion| "V" - Value
 - Reprezinta tipul coloanei
 - cCol
 - Obligatoriu
 - Tip: Character sau "C"|"N"|"D"|"M"|"L" daca cColTip = "V"
 - Valori :
 - expresia de evaluat in cazul cColTip := "E"
 - nume camp in cazul cColTip := "F"
 - valoare constanta din coloana respectiva
 - cColType
 - Obligatoriu in cazul cColTip := "E" sau "V"
 - Tip: character
 - Valori posibile : "C"|"N"|"D"|"M"|"L"
 - in cazul cColTip = "F" - Valoarea := FILEDTYPE(cCol)
 - reprezinta tipul campului de exportat
 - nColLenght
 - Obligatoriu in cazul cColTip := "E" sau "V"
 - Tip : Numeric
 - Valori posibile: numere intregi > 0
 - reprezinta lungimea maxima a campului exportat in caractere
 - in cazul cColTip="F" si valori incorecte se inlocuieste cu FIELDLEN(cCol) sau 50 in cazul campurilor Memo
 - nColDec
 - Obligatoriu in cazul cColTip="E" sau "V"

- Valori posibile: numere intregi ≥ 0 si $\leq nColLenght$
- Reprezinta numarul zecimalelor in numar de caractere
- in cazul cColTip="F" si valori incorecte se inlocuieste cu FIELDDEC(cCol)
- cColPict
 - Tip: Character
 - Valori posibile: NIL|<sr>
 - Valoare implicita :
 - "@" pentru cColType = "C"|"L"|"M"
 - "Short Date" pentru cColType = "D"
 - '#,##0'+IIF(nColDec>0,','+ REPLICATE('0',nColDec),'') pentru cColType "N"
 - Reprezinta formatarea celulei
- lColWrap
 - Tip: logical
 - Valori posibile NIL, .t., .f.
 - Valoare implicita : .f.
 - Reprezinta daca se activeaza optiunea "Wrap" a celulei
 - In cazul campurilor de tip Memo se reseteaza la .t.
- textul din liniile 2,3,... se vede doar la dublu-click pe prima linie
- gColTotal
 - Tip: logical sau text
 - Valori posibile: NIL, Text, .t., .f.
 - Valoare implicita .f.
 - Reprezinta daca se introduce o formula de total pentru coloana respectiva.
- In cazul in care valoarea este de tip "C" atunci textul respectiv se va inscrie in randul de total.
- Fiecare element reprezinta o Coloana
- pc_File
 - Obligatori
 - Tip: Caracter
 - In cazul pc_TipOp = ">"|"-" poate fi si NIL
 - Valoarea Reprezinta Fisierul XML in care se va efectua exportul
- @pc_Error
 - Obligatori
 - Tip: Caracter
 - In aceasta variabila se stoceaza mesajul de eroare
- pc_CP
 - Optional
 - Tip: Caracter
 - Reprezinta CodePageul, prin care se face transformarea textelor in format html
 - Valori Posibile "852","437","1250",EMPTY,NIL
 - Daca valoarea parametrului este NIL sau empty valoarea implicita este "852"

Returnează

Reurnează .T. dacă operația a fost realizată sau .F. în caz de eșec și completează pc_Error

Funcționare

Funcția poate fi apelată în două moduri, și anume:

cu opțiunea pc_TipOp:=""

În acest caz se va genera fișierul pentru toate înregistrările din zona curentă de lucru

cu opțiunile pc_TipOp:= "<" - pt. a scrie începutul fișierului; "-" pentru fiecare rând în parte și ">" pentru închidere fișier

Limitarea lățimii unei coloane este de 400 de puncte (cca. 60 caractere).

Limitarea înălțimii unui rând este de 75 de puncte (cca. 5 rânduri).

În cazul antetului, textul poate să conțină OSNewline().

Exemplul 1:

```
Coloana1 := {"NUME"}, "F", "USER",,,,,, "Total:"}
Coloana2 := {{"}, "F", "NAME",,,,,,.f.}
Coloana3 := {"PACK"}, "F", "PACK",,,4,0,,,,.f.}
Coloana4 := {"LAST", "PACK",}, "F", "LASTPACK", "D",10,0,["$-408]d\ mmmm\ yyyy;@",,.f.}
Coloana5 := {"Data", "Exportului", "DATE()", "E", "DATE()", "D",12,0,["$-40E]yyyy/\ mmm/\ d\.;@",.f.,.f.}
Coloana6 := {"Nr.", "Inreg.", "RECNO()", "E", "RECNO()", "N",7,0, "0",.f.,.t.}
Coloana7 := {"Ctrl.", "Inreg.", "E", "RECCOUNT()-RECNO()+1", "N",7,0, "0",.f.,.t.}
Coloana8 := {"Ctrl.", "PACK", "E", "IIF(FILES->PACK=.t.,.f.,.t.)", "L",5,0,,,,.f.}
Coloane := {Coloana1,Coloana2,Coloana3,Coloana4,Coloana5,Coloana6,Coloana7,Coloana8}
Antet1 := {"Rand antet 1 , aliniat stanga", "L"}
Antet2 := {"Rand antet 2 , Aliniere neprecizata",}
Antet3 := {,,} //rand gol
Antet4 := {"Rand antet 4 , Aliniere Centru", "C"}
Antet := {Antet1,Antet2,Antet3,Antet4}
mesaj := ""
USE FILES ALIAS FILES
IF h_ExpXlsXml(
    '',';
    Antet,;
    Coloane,;
    'Fisier.xml',;
    @mesaj)
ELSE
    ? mesaj
ENDIF

lRetErr := h_ExpXlsXml("<",Antet,Coloane, 'Fisier.xml',@mesaj)
nRecno := RECNO()
DBGOTOP()
DO WHILE .NOT. EOF() .AND. lRetErr
    lRetErr := h_ExpXlsXml("-",,,,@mesaj)
    DBSKIP()
ENDDO
```

```

lRetErr := lRetErr .and. h_ExpXlsXml(">",,,,@mesaj)
DBGOTO(nRecno)
IF !lRetErr
    ?mesaj
ENDIF

```

Exemplul 2:

```

Antet := {}
AADD(Antet, {"Rand antet 1 , aliniat stanga","L"})
AADD(Antet, {"Rand antet 2 , Aliniere neprecizata",})
AADD(Antet, {},) //rand gol
AADD(Antet, {"Titlu, Aliniere Centru","C"})
AADD(Antet, {},) //rand gol

```

```

Coloane := 0
Coloana1 := {"NUME"},"F","USER",,,,,,"Total:"}
Coloana2 := {{"F"},"NAME",,,,,,.f.}
Coloana3 := {"PACK"},"F","PACK",,4,0,,,f.}
Coloana4 := {"LAST"},"PACK",,,"F","LASTPACK","D",10,0,"[$-408]d\ mmmm\ yyyy;@",,,f.}
Coloana5 := {"Data"},"Exportului","DATE()"},"E","DATE()","D",12,0,"[$-40E]yyyy/\ mmm/\ d\.;@",.f.,.f.}
Coloana6 := {"Nr."},"Inreg."},"RECNO()"},"E","RECNO()","N",7,0,"0",.f.,.t.}
Coloana7 := {"Ctrl."},"Inreg."},"E","RECCOUNT()-RECNO()+1","N",7,0,"0",.f.,.t.}
Coloana8 := {"Ctrl."},"PACK"},"E","IIF(FILES->PACK=.t.,.f.,.t.)"},"L",5,0,,,f.}

```

```

AADD(Coloane,{{"NUME"},"F","USER",,,,,,"Total:"})
AADD(Coloane,{{"F"},"NAME",,,,,,.f.})
AADD(Coloane,{{"PACK"},"F","PACK",,4,0,,,f.})
AADD(Coloane,{{"LAST"},"PACK",,,"F","LASTPACK","D",10,0,"[$-408]d\ mmmm\ yyyy;@",,,f.})
AADD(Coloane,{{"Data"},"Exportului","DATE()"},"E","DATE()","D",12,0,"[$-40E]yyyy/\ mmm/\ d\.;@",.f.,.f.})
AADD(Coloane,{{"Nr."},"Inreg."},"RECNO()"},"E","RECNO()","N",7,0,"0",.f.,.t.})
AADD(Coloane,{{"Ctrl."},"Inreg."},"E","RECCOUNT()-RECNO()+1","N",7,0,"0",.f.,.t.})
AADD(Coloane,{{"Ctrl."},"PACK"},"E","IIF(FILES->PACK=.t.,.f.,.t.)"},"L",5,0,,,f.})

```

Exemplul 3 de gestionare a fişierului XML:

```

xx_fis := h_TruePath(h_AddPath("FISEC_"+h_StatId()+".XML",h_FilePathes[fil_tmp]))
#IF !h_SafeErase(xx_fis)
    Abandon()
#ENDIF
...
#IF !lRetErr
    h_MesErr(xx_mesaj,0)

```

```
#ELSE
    X_1 := xx_fis
    sScr := gcmd_Getcmd("RUND")
    #SDO sScr
#ENDIF
```

Exemplul 4:

```
xx_Antet := {}
AADD(xx_Antet, {x_SocData()+"", CIF:"+x_SocData("COD_FISCAL"), "L"})
AADD(xx_Antet, {"Data:"+h_DTOC(DATE()), "zz/ll/aaaa"), "L"})
AADD(xx_Antet, {"Incasari/Plati "+h_DTOC(pp_inc, "zz/ll/aaaa")+h_DTOC(pp_sf, "-zz/ll/aaaa"), "C"})
AADD(xx_Antet, {}, {})

xx_Col := {}
AADD(xx_col, {"Jurnal"}, "F", "NRJUR", , , , , "Total:"))
AADD(xx_col, {"Numar", "fila"}, "F", "NRFILA", , , , , .f.})
AADD(xx_col, {"Data", "fila"}, "F", "DATA", , , , , .f.})
AADD(xx_col, {"Tip", "inc/pla"}, "F", "TIPIP", , , , , .f.})
AADD(xx_col, {"Tip", "docum"}, "F", "TIPDOC", , , , , .f.})
AADD(xx_col, {"Numar", "docum"}, "F", "NRDOC", , , , , .f.})
AADD(xx_col, {"Data", "document"}, "F", "DATADOC", , , , , .f.})
AADD(xx_col, {"NrFac", "NrRec"}, "F", "NRFAC", , , , , .f.})
AADD(xx_col, {"Observatie 1", ""}, "F", "OBS1FAC", , , , , .f.})
AADD(xx_col, {"Observatie 2", ""}, "F", "OBS2FAC", , , , , .f.})
AADD(xx_col, {"Cod", "partener"}, "F", "PARTENER", , , , , .f.})
AADD(xx_col, {"Denumire/Obs", "partener"}, "F", "OBSPART", , , , , .f.})
AADD(xx_col, {"Cont", "corespondent"}, "F", "CONTCR", , , , , .f.})
AADD(xx_col, {"Sume", "incasate"}, "E", "IF(inc->TIPIP='I', inc->SUMA, 0)", "N", 15, 2, , , .t.})
AADD(xx_col, {"Sume", "platite"}, "E", "IF(inc->TIPIP='P', inc->SUMA, 0)", "N", 15, 2, , , .t.})
AADD(xx_col, {"Sold"}, "E", "IF(inc->TIPIP='S', inc->SUMA, 0)", "N", 15, 2, , , .f.})

#IF xx_mon
    AADD(xx_col, {"Moneda"}, "F", "MONEDA", , , , , })
    AADD(xx_col, {"Curs", "schimb"}, "E", "csch_calcul(inc->SUMAVAL, inc->SUMA)", "N", 9, 6, , , })
    AADD(xx_col, {"Sume", "incasate", "(valuta)"}, "E", "IF(inc->TIPIP='I', inc->SUMAVAL, 0)", "N", 12, 2, , , })
    AADD(xx_col, {"Sume", "platite", "(valuta)"}, "E", "IF(inc->TIPIP='P', inc->SUMAVAL, 0)", "N", 12, 2, , , })
    AADD(xx_col, {"Sold", "", "(valuta)"}, "E", "IF(inc->TIPIP='S', inc->SUMAVAL, 0)", "N", 12, 2, , , })
    //AADD(xx_col, {"Suma", "factura"}, "F", "SUMAFAC", , , , , })
    //AADD(xx_col, {"Suma", "reevalua"}, "F", "SUMAREV", , , , , })
#ENDIF
AADD(xx_col, {"Mesaj eroare"}, "E", "xx_xerr", "C", 100, 0, , , })
```

CODURI DE EROARE

```
100 - PCOUNT() < 5 .OR. VALTYPE(pc_TipOp)<>"C" .OR. .NOT. (ALLTRIM(pc_TipOp) IN {"", "<", ">", "-"})
    - numar incorrect de parametrii
    - pc_TipOp nu este in {" ", "", "<", ">", "-"}
101 - EMPTY(pc_File) .OR. VALTYPE(pc_File)<>"C"
    - nume pc_File gol sau nu este de tip caracter
102 - VALTYPE(pa_Antet) <>"A"
    - tip pa_Antet nu este de tip matrice
103 - VALTYPE(pa_Antet[i])<>"A" .OR. LEN(pa_Antet[i])<2
    - element pa_Antet nu este de tip array sau numar elemente incorrect
104 - EMPTY(pa_Camp) .OR. VALTYPE(pa_Camp)<>"A"
    - pa_Camp null sau gol sau nu este de tip array
105 - LEN(pa_Camp[i])<8
    - lungime elemente pa_Camp incorrect
106 - VALTYPE(pa_Camp[i,1])<>"A"
    - aCapTabel nu este de tip array
107 - EMPTY(pa_Camp[i,2]) .OR. VALTYPE(pa_Camp[i,2])<>"C" .OR. .NOT.(pa_Camp[i,2] IN {"F", "E", "V"})
    - cColTip nu este "F"/"E"/"V"
108 - EMPTY(pa_Camp[i,3]) .OR. VALTYPE(pa_Camp[i,3])<>"C"
    - nume coloana sau expresie nu este precizat sau nu este de tip caracter - exceptie daca cColTip este "V"
109 - (pa_Camp[i,2] = "E" .OR. pa_Camp[i,2] = "V") .AND. ( EMPTY(pa_Camp[i,4]) .OR. VALTYPE(pa_Camp[i,4])<>"C" .OR. .NOT.
    (pa_Camp[i,4] IN {"D", "C", "N", "L", "M"}))
    - cColTip = "E"/"V" si cColDtype incorrect
110 - (pa_Camp[i,2] = "E" .OR. pa_Camp[i,2] = "V") .AND. ( VALTYPE(pa_Camp[i,5])<>"N" .OR. pa_Camp[i,5]<=0)
    - cColTip = "E"/"V" si nColLenght nu este numeric sau este mai mic sau egal cu 0
111 - (pa_Camp[i,2] = "E" .OR. pa_Camp[i,2] = "V") .AND. (VALTYPE(pa_Camp[i,6])<>"N" .OR. INT(pa_Camp[i,5])<INT(pa_Camp[i,6]))
    - cColTip = "E"/"V" sau "V" si nColDec nu este numeric sau nColLenght<nColDec
112 - pa_Camp[i,2] = "F" .AND. (FIELDPOS(pa_Camp[i,3])=0)
    - cColTip = "F" si nu exista campul cCol
113 - pc_Cp nu corespunde specificatiilor
    - pc_CP trebuie sa fie NIL sau caracter cu valorile empty, "852", "437", "1250"
130 - VALTYPE(<field>|<expresie>) <> aCamp[i,4] - exceptie <expresie> este "C" si aCamp[i,4] = "M"
    - Valtype cCol difera de cel precizat in cColDtype
140 - eroare secventa de apel h_ExpXlsXml()
    implementat cu err_secvapel()
201 - FILE(pc_File) .AND. FERASE(pc_File) = -1
    - Nu s-a reusit stergerea fisierului pc_File
202 - FCREATE(pc_File) = -1
    - Nu s-a reusit crearea pc_File
203 - FCLOSE()
    - Nu s-a reusit inchiderea pc_File
300 - FWRITE()
    - Nu s-a reusit scrierea in pc_File
```

2. FUNCȚIA h_Dbf2XlsXml

Exportă un set de date în format xml Spreadsheet

Parametri:

- pc_File
 - Obligatoriu
 - Tip: Caracter
 - In cazul pc_TipOp = ">"|"-" poate fi si NIL
 - Valoarea Reprezinta Fisierul XML in care se va efectua exportul
- pg_Alias
 - Optional
 - Tip caracter sau numeric
 - Contine numele aliasului sau zona de lucru a tabeli de exportat
- pa_Col
 - Optional
 - Tip Array
 - Forma : {"Nume Coloana1", [...[Nume Coloana n]]}
 - Contine lista coloanelor aferente aliasului de exportat
 - Daca este {} sau NIL se exporta toate coloanele din tabela
- @pc_Error
 - Obligatoriu
 - Tip: Caracter
 - In aceasta variabila se stocheaza mesajul de eroare
- pc_CP
 - Optional
 - Tip: Caracter
 - Reprezinta CodePageul, prin care se face transformarea textelor in format html
 - Valori Posibile "852", "437", "1250", EMPTY, NIL
 - Daca valoarea parametrului este NIL sau empty valoarea implicita este "852"

Returneaza

Returneaza .T. daca operatia a fost realizata sau .F. in caz de esec si completeaza pc_Error

Functionare

Functia alege tabela precizata prin pg_Alias(numar sau nume),
Construieste Array-ul de pa_Camp in functie de structura tabeli si a elementelor din pa_Col
Si apeleaza functia h_ExpXlsXml

Coduri de Eroare:

In cazul in care codul este returnat de h_ExpXlsXml, codurile sunt la fel ca si la descrierea functiei h_ExpXlsXml.

Coduri de eroare proprii:

- 1 - PCOUNT()<>4
Numar incorrect de parametrii
- 2 - EMPTY(pc_File) .OR. VALTYPE(pc_File)<>"C"
Nume fisier incorrect
- 3 - .NOT. (VALTYPE(pg_Alias)<>"C" .OR. VALTYPE(pg_Alias)<>"N")
pg_Alias nu este de tip numeric sau caracter
- 4 - Aliasul sau zona de lucru precizata este incorecta
- 5 - .NOT. EMPTY(pa_Col) .AND. VALTYPE(pa_Col)<>"A"
pa_Col nu este gol si nu este de tip array
- 6 - EMPTY(pa_Col[i]) .OR. VALTYPE(pa_Col[i])<>"C"
Elementele pa_Col sunt goale sau nu sunt de tip caracter
- 7 - Elemente din pa_Col nu se gasesc in campurile tabeli
- 8 - Parametrul pc_CP nu corespunde specificatiilor

Exemple:

h_Dbf2XlsXml("fisier.xml",2,,@mesaj)

sau

h_Dbf2XlsXml("fisier.xml","Alias",{"coloana1","coloana2"},@mesaj)