HDFTACTGRP(\*NO) ACTGRP(\*new)

HDECEDIT('.')

\*------------------------------------------------------------------

FLFFIRM UF E K DISK EXTFILE('ODIVERS/LFFIRM')

FJNLFNO IF E K DISK

FLFMVTKA UF E K DISK EXTFILE('TESTKOEN/LFMVTKA')

FLFMVTK3 IF E K DISK EXTFILE('TESTKOEN/LFMVTK3')

F RENAME(PFMVTKR:MVTKR) PREFIX

FD365LOGF O A E DISK

\*-------------------------------------------------------------------

D xmlHandler PR 10i 0

D ignore 1a

D event 10i 0 value

D string \* value

D stringLen 20i 0 value

D exceptionId 10i 0 value

\*-------------------------------------------------------------------

D XML s 500a varying

D ignoreMe s 1a

D xmlfile s 256a

D pdf s 256a

D firma s 3a

D library s 10a

D klant s 8a

D klnr s 5s 0

D faknr s 7s 0

D voucher s 20a

D transdate s 10a

D datclc s d

D dat8 s 8s 0

D datum s 8a

D sqlstmt s 500a

D amt s 11s 2

D fil1 s 21a

D fil2 s 21a

D wkArray s 10a dim(5)

D d s 10i 0

D jnAry s 3a dim(10)

D dcAry s 7s 0 dim(10)

D vcAry s 20a dim(10)

D ijAry s 4a dim(10)

D imAry s 2a dim(10)

D idAry s 2a dim(10)

D journaal s 10a

D txt s 30a

D txtary s 20a dim(10)

\*----------------------------------------------------------------

D DS

DJJ 1 4 0

DMM 5 6 0

DDD 7 8 0

DIMPD 1 8 0

----------------------------------------------------------------

C \*entry plist

C parm xmlfile 256

\*----------------------------------------------------------------

C/EXEC SQL

C+ SET OPTION COMMIT=\*NONE

C/END-EXEC

\*----------------------------------------------------------------

C eval XML = %trimr(xmlfile)

C monitor

C xml-sax %handler(xmlHandler: ignoreMe)

C %XML(%trim(XML): 'doc=file')

C on-error

C endmon

\*----------------------------------------------------------------

C eval \*inlr = \*on

\*----------------------------------------------------------------

P xmlHandler B

D xmlHandler PI 10i 0

D ignore 1a

D event 10i 0 value

D string \* value

D stringLen 20i 0 value

D exceptionId 10i 0 value

D value s 65535a based(String)

D ucs2val s 16363c based(String)

D MAX\_DEPTH c 100

D depth s 10i 0 inz(0) static

D stackname s 65535a varying inz('')

D dim(MAX\_DEPTH)

D static

D stackval s 65535a varying inz('')

D dim(MAX\_DEPTH)

D static

C select

\* start XML document intialisatie

C when event = \*XML\_START\_DOCUMENT

C eval depth=0

C eval stackname(\*) = ''

C eval stackval(\*) = ''

\* meer elementen dan 64= fout in parsing

C when depth >= MAX\_DEPTH

C and ( event = \*XML\_START\_ELEMENT

C or event = \*XML\_ATTR\_NAME )

C return -1

\* start van nieuw element

C when event = \*XML\_START\_ELEMENT

C eval depth += 1

C if depth = 1

C eval stackname(depth) = '/'

C + %subst(value:1:stringLen)

C else

C eval stackname(depth) = stackname(depth-1

C + %subst(value:1:stringLen)

C endif

C eval stackval(depth) = ''

\* start van nieuw attribut

C when event = \*XML\_ATTR\_NAME

C eval depth += 1

C eval stackname(depth) = stackname(depth-1

C + %subst(value:1:stringLen)

C eval stackval(depth) = ''

\* start van nieuw attribut

C when event = \*XML\_END\_ELEMENT

C or event = \*XML\_END\_ATTR

C if depth > 0

C select

C when stackname(depth) =

C '/OptiGrator/CustTrans/dataAreaId'

C eval library=\*blanks

C eval firma=\*blanks

C eval firma=%trim(stackval(depth))

C select

C when firma = 'wnv'

C eval library = 'WVDJ'

C z-add 40 fnr 3 0

C when firma = 'act'

C eval library = 'ACDJ'

C z-add 20 fnr

C when firma = 'grp'

C eval library = 'WGDJ'

C z-add 1 fnr

C when firma = 'hls'

C eval library = 'HEDJ'

C z-add 31 fnr

C when firma = 'ind'

C eval library = 'INDJ'

C z-add 35 fnr

C when firma = 'int'

C eval library = 'WIDJ'

C z-add 2 fnr

C when firma = 'sfo'

C eval library = 'SFDJ'

C z-add 64 fnr

C when firma = 'wbl'

C eval library = 'BADJ'

C z-add 22 fnr

C endsl

C eval library='TESTKOEN'

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'CustTable/ALTLegacyCustID '

C eval klant=\*blanks

C eval klant=%trim(stackval(depth))

C eval klnr=%dec(klant:5:0)

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'Invoice'

C eval faknr=%dec(%trim(stackval(depth)):7:0)

C when stackname(depth) = '/OptiGrator/CustTrans/'

C +'CustSettlement/custtrans.Transdate-display'

C eval txt=\*blanks

C eval txt=%trim(%subst(stackval(depth):1:10))

C if d <> 0

C EVAL IJARY(d)= %SUBST(TXT:7:4)

C EVAL IMARY(d)= %SUBST(TXT:4:2)

C EVAL IDARY(d)= %SUBST(TXT:1:2)

C endif

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'CustSettlement/custTrans.Voucher-display'

C eval voucher=\*blanks

C eval voucher=%trim(stackval(depth))

C eval wkarray=%split(voucher:'-')

\*\* als voucher geen koppelteken heeft

C if %subst(voucher:1:2) = 'DI'

C or %subst(voucher:1:2) = 'OP'

C if %scan('-':voucher) = 0

C eval wkarray(1) = %subst(voucher:1:3)

C eval wkarray(2) = %subst(voucher:4:7)

C endif

C endif

\* factuur of betaling ?

C IF WKARRAY(2) = \*BLANKS

C add 1 d

C MOVEL '001' JNARY(d)

C EVAL DCARY(d)= %DEC(%TRIM(WKARRAY(1)):7:0)

C ELSE

C KEYFNO KLIST

C KFLD FIRMA

C KFLD JNLFO

C MOVEL WKARRAY(1) JNLFO

C add 1 d

C eval vcary(d) = voucher

C KEYFNO CHAIN JNLFNO

C IF %FOUND

C MOVEL JNL400 JNARY(d)

C ELSE

C MOVEL '030' JNARY(d)

C ENDIF

C EVAL DCARY(d)= %DEC(%TRIM(WKARRAY(2)):7:0)

C ENDIF

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'CustSettlement/custTrans.'

C + 'altDisplayJournal-display'

C eval journaal=\*blanks

C eval journaal=%trim(stackval(depth))

C d ifne 0

C MOVEL journaal JNLFO

C add 1 d

C KEYFNO CHAIN JNLFNO

C IF %FOUND

C MOVEL JNL400 JNARY(d)

C ENDIF

C ENDIF

\*\*

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'CustSettlement/custTrans.Txt-display'

C eval txt=\*blanks

C eval txt=%trim(%subst(stackval(depth):1:10))

C if %scan('\_':%trim(txt)) > 0

C if txt <> \*blanks

C eval txtary=%split(txt:'\_')

C d ifne 0

C EVAL JNARY(d)= %TRIM(WKARRAY(1))

C EVAL DCARY(d)= %DEC(%TRIM(WKARRAY(2)):7:0)

C EVAL IJARY(d)= %SUBST(WKARRAY(3):7:4)

C EVAL IMARY(d)= %SUBST(WKARRAY(3):4:2)

C EVAL IDARY(d)= %SUBST(WKARRAY(3):1:2)

C endif

C endif

C endif

\*\*

C when stackname(depth) = '/OptiGrator/CustTrans/'

C + 'CustSettlement/createdDateTime'

C eval transdate=\*blanks

C eval transdate=%trim(%subst(stackval(depth):1:10))

C monitor

C eval datclc = %date(%trim(transdate):\*usa)

C on-error

C eval datclc = %date()

C endmon

C \*iso movel datclc dat8

C movel dat8 datum

C when stackname(depth) = '/OptiGrator/CustTrans'

C monitor

C eval library='TESTKOEN'

// clearen aanzuiveringsinfo

C if faknr <> 0

C eval fil1 = %trim(library) + '/LFMVTKA'

C eval fil2 = %trim(library) + '/LFMVTK3'

C open lfmvtka

C open lfmvtk3

C faknr chain lfmvtk3

C if %found

C keyaan klist

C kfld XMKLNR

C kfld XMNAANZ

C xmnaanz ifne 0

C keyaan setll lfmvtka 5555

C keyaan reade lfmvtka 55

C \*in55 doweq \*off

C z-add 0 MNAANZ

C movel \*blanks MKAANZ

C update pfmvtkr

C keyaan reade lfmvtka 55

C enddo

C endif

C endif

C close lfmvtka

C close lfmvtk3

C endif

// ophalen nieuw nummer

C FNR CHAIN LFFIRM 55

C ADD 1 FINRAN

C UPDATE PFFIRMR

C Z-ADD FINRAN AANZNR 7 0

// opvullen nummers

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MJN = 1 '

C + ' AND MFAKNR = '

C + %TRIM(%EDITC(FAKNR:'4'))

C/EXEC SQL

C+ EXECUTE IMMEDIATE :SQLSTMT

C/END-EXEC

C if (SQLCOD < 0)

C eval msg= 'NIET alle aanzuiveringen van bestand '

C + %trimr(xmlfile) + ' werden verwerkt.'

C call 'D365IN75M'

C parm msg 1024

C parm msge 1

C time logtime

C write D365LOGR

C endif

C 1 do d z 2 0

C if jnary(z) = '001'

C ijary(z) ifne \*blanks

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MJN = 1'

C + ' AND MFAKNR = '

C + %TRIM(%EDITC(DCARY(z):'4'))

C + ' AND MIMPJJ = '+%TRIM(IJARY(z))

C + ' AND MIMPMM = '+%TRIM(IMARY(z))

C + ' AND MIMPDD = '+%TRIM(IDARY(z))

C else

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MJN = 1'

C + ' AND MFAKNR = '

C + %TRIM(%EDITC(DCARY(z):'4'))

C endif

C else

C vcary(z) ifne \*blanks

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ', MFAKNR = '

C + %TRIM(%EDITC(FAKNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MOPM = ' + %trim(vcary(z))

C else

C ijary(z) ifne \*blanks

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ', MFAKNR = '

C + %TRIM(%EDITC(FAKNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MJN = ' + %trim(jnary(z))

C + ' AND MNRDOK = '

C + %TRIM(%EDITC(DCARY(z):'4'))

C + ' AND MIMPJJ = '+%TRIM(IJARY(z))

C + ' AND MIMPMM = '+%TRIM(IMARY(z))

C + ' AND MIMPDD = '+%TRIM(IDARY(z))

C else

C eval sqlstmt = 'UPDATE ' +%trim(library) + '/'

C + 'PFMVTK SET MNAANZ = '

C + %TRIM(%EDITC(AANZNR:'4'))

C + ', MFAKNR = '

C + %TRIM(%EDITC(FAKNR:'4'))

C + ' WHERE MKLNR = '

C + %TRIM(%EDITC(KLNR:'4'))

C + ' AND MJN = ' + %trim(jnary(z))

C + ' AND MNRDOK = '

C + %TRIM(%EDITC(DCARY(z):'4'))

C endif

C endif

C endif

C/EXEC SQL

C+ EXECUTE IMMEDIATE :SQLSTMT

C/END-EXEC

C if (SQLCOD < 0)

C eval msg= 'NIET alle aanzuiveringen van bestand '

C + %trimr(xmlfile) + ' werden verwerkt.'

C call 'D365IN75M'

C parm msg 1024

C parm msge 1

C time logtime

C write D365LOGR

C endif

C enddo

// afpunten op nummer

C on-error

C endmon

C eval d=0

C eval dcary=0

C eval vcary=\*blanks

C eval jnary=\*blanks

C endsl

C eval depth -= 1

c endif

C when event = \*XML\_CHARS

C or event = \*XML\_PREDEF\_REF

C or event = \*XML\_ATTR\_CHARS

C or event = \*XML\_ATTR\_PREDEF\_REF

C eval stackval(depth) += %subst(value:1:stringLen)

\* omzetten van utf

C when event = \*XML\_UCS2\_REF

C or event = \*XML\_ATTR\_UCS2\_REF

C eval stackval(depth) += %char(%subst( ucs2val : 1

C : %div(stringLen:2) ))

C endsl

C return 0

P E