```
SOURCE FILE . . . . . . .  CDJSON/QRPGLESRC
MEMBER  . . . . . . . . .  JSONINV01

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100 **FREE
  200   Ctl-Opt BndDir('JSONXML') dftactgrp(*NO) ACTGRP('QILE');
  300   DCL-F lffakinv DISK(*EXT) USAGE(*INPUT) KEYED;
  400   DCL-F lfstafaf DISK(*EXT) USAGE(*INPUT) KEYED;
  500   DCL-F lfbew2a  DISK(*EXT) USAGE(*INPUT) KEYED;
  600   DCL-F pfsigi   DISK(*EXT) USAGE(*INPUT) KEYED;
  700   DCL-F lforderb DISK(*EXT) USAGE(*INPUT) KEYED;
  800   DCL-F pfverkop DISK(*EXT) USAGE(*INPUT) KEYED;
  900   DCL-F LFPROJFFA1 DISK(*EXT) USAGE(*INPUT) KEYED;
 1000   DCL-F lfvprs DISK(*EXT) USAGE(*INPUT) KEYED;
 1100   DCL-F lfklan DISK(*EXT) USAGE(*INPUT) KEYED;
 1200   DCL-F lfvern DISK(*EXT) USAGE(*INPUT) KEYED;
 1300   DCL-F pffactur DISK(*EXT) USAGE(*INPUT) KEYED;
 1400   DCL-F vatlogf DISK(*EXT) USAGE(*UPDATE:*OUTPUT) KEYED;
 1500   DCL-F LFJSONINV DISK(*EXT) USAGE(*UPDATE:*OUTPUT) KEYED;                                        27/11/23
 1600   /include qrpgleRef,jsonxml
 1700
 1800    dcl-pi *n ;
 1900      pfirma char(2) ;
 2000      pdatum char(8) ;
 2100    end-pi ;
 2200
 2300   dcl-s pklr char(1) ;
 2400
 2500   Dcl-S klnr zoned(8:0);
 2600   Dcl-S kbase zoned(9:2);
 2700   Dcl-S teller int(10);
 2800   Dcl-S vatregime char(1);
 2900   Dcl-S pTemp              Pointer;
 3000   Dcl-S pFakn              Pointer;
 3100   Dcl-S pJson              Pointer;
 3200   Dcl-S pNewObj            Pointer;
 3300   Dcl-S werk zoned(11:4);
 3400   Dcl-S prc zoned(3:1);
 3401   Dcl-S FILENAME char(30);                                                                        01/12/23
 3402   FILENAME = %trim(%char(%timestamp(*sys:0)));                                                    01/12/23
 3500
 3600   pklr = 'K';
 3700
 3800   kimpda = %dec(pdatum:8:0);
 3900
 4000   pNewObj = json_NewObject();
 4100
 4200   setll (kimpda) lffakinv;
 4300   reade (kimpda) lffakinv;
 4400   dow not %eof(lffakinv);
 4500   chain(n) (pfirma:KFAKNR) lfjsoninv;                                                             27/11/23
 4600   if not %found;                                                                                  27/11/23
 4700   chain ('001':KFAKNR) lfbew2a;
 4800   //if bbufla = '';                                                                               27/11/23
 4900   if bklr = 'K';
 5000   teller = teller + 1;
 5100   pJson = json_NewObject();
```

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
chain (pklr:bnum3) pfsigi;
json_SetStr    (pJson : 'companyId'  : pfirma );
json_SetInt    (pJson : 'invoiceNumber'  : kfaknr );
//json_SetStr    (pJson : 'voucherNumber'  : bjnl + '/' + %char(bdocnr) );
json_SetStr    (pJson : 'voucherNumber'  :  %char(bdocnr) );
if kbdrf < 0;                                                                                                     16/11/23
baard = 'CN';
kbdrf = kbdrf * -1;
kbtw = kbtw * -1;                                                                                                 22/11/23
kegal = kegal * -1;
kbdr0 = kbdr0 * -1;
kbdr6 = kbdr6 * -1;                                                                                               16/11/23
kbdr17 = kbdr17 * -1;
kbdr19 = kbdr19 * -1;
kbdr25 = kbdr25 * -1;
kbdr33 = kbdr33 * -1;
kbdr0m = kbdr0m * -1;
KBD195 = KBD195 * -1;
KBDR12 = kbdr12 * -1;
kkortb = kkortb * -1;
endif;

json_SetStr    (pJson : 'type'  : %trim(baard) );
klnr = %dec(bnum:8:0);
monitor;
kklnr = klnr;
on-error;
kklnr = 0;
endmon;
json_SetStr    (pJson : 'customerId'  : pfirma + bnum3);
json_SetNum    (pJson : 'cashDiscount'  : kkortb );
// herrekenen perc disc op vraag Sebastien
prc = 0;
if kkortb <> 0;
   werk = kbdrf - kbtw;
   werk = werk / 100;
if werk <> 0;
   prc = %dech((kkortb / werk): 3 :1);
endif;
endif;
json_SetNum    (pJson : 'CashDiscountPercentage'  : prc );
json_SetStr    (pJson : 'currency'  : 'EUR' );
json_SetNum    (pJson : 'totalInvoiceAmount'  : kbdrf );
chain (pfirma:KFAKNR) pffactur;
if %found;
json_SetStr    (pJson : 'OGM': %trim(KOGM));
else;
json_SetStr    (pJson : 'OGM': '');
endif;
if bvald = *blanks;
eval bvald = bimpd;
endif;
json_SetStr    (pJson : 'invoiceDate'  : %subst(bimpd:1:4) + '-' +
                %subst(bimpd:5:2) + '-' + %subst(bimpd:7:2));
json_SetStr    (pJson : 'dueDate'  : %subst(bvald:1:4) + '-' +
                %subst(bvald:5:2) + '-' + %subst(bvald:7:2));
json_SetNum    (pJson : 'totalVatAmount'  : kbtw );
kbase = kbdrf - kbtw - kkortb;
json_SetNum    (pJson : 'totalBaseAmount'  : kbase );
json_SetStr    (pJson : 'vatCode'  : bbtwc );
json_SetStr    (pJson : 'description'  : boms );
json_SetStr    (pJson : 'invoiceCountry'  : %trimr(LANDK));
json_SetStr    (pJson : 'deliveryCountry'  : %trimr(LANDK));
json_SetInt    (pJson : 'orderNumber'  : bordnr);

// check VAT regime
// G = IC goederen
// E = Export
// S = IC diensten
// M = medecontractant
vatregime = *blanks;
chain (bjnl:bdocnr:bimpd:bklr3:bnum3:'R':'700014') lfbew2a;
if %found;
vatregime = 'S';
endif;
chain (bjnl:bdocnr:bimpd:bklr3:bnum3:'R':'700015') lfbew2a;
if %found;
vatregime = 'M';
endif;
chain (bjnl:bdocnr:bimpd:bklr3:bnum3:'R':'700016') lfbew2a;
if %found;
vatregime = 'G';
endif;
chain (bjnl:bdocnr:bimpd:bklr3:bnum3:'R':'700017') lfbew2a;
if %found;
vatregime = 'E';
endif;
chain (bjnl:bdocnr:bimpd:bklr3:bnum3:'R':'700110') lfbew2a;
if %found;
vatregime = 'G';
endif;

if vatregime = *blanks;
if pfirma = 'WV';
chain (bdocnr) PFVERKOP;
if %found;
vatregime = 'E';
endif;
endif;
if kbdr0m <> 0;
vatregime = 'M';
endif;
endif;

if kbdrf = 0;
vatregime = 'Z';
```

```
SOURCE FILE . . . . . . .  CDJSON/QRPGLESRC
MEMBER  . . . . . . . . .  JSONINV01

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
        endif;

        json_SetStr    (pJson : 'vatRegime' : %trimr(vatregime));

        chain (pfirma:bdocnr) vatlogf;
        if %found;
        vregime = vatregime;
        update vatr;
        else;
        vfirma = pfirma;
        vdocnr = bdocnr;
        vregime = vatregime;
        write vatr;
        endif;

        chain bordnr lforderb;
        if %found;
        json_SetStr    (pJson : 'customerReference' : %trimr(OREFK));
        json_SetInt    (pJson : 'articleId'  : oartnr);
        json_SetStr    (pJson : 'articleDescription' : %trimr(OARTUI));
        json_SetInt    (pJson : 'quantity'  : ostuks);
        json_SetNum    (pJson : 'articlePrice'  : ofaktw);
        json_SetNum    (pJson : 'installationPrice'  : owerpl);
        else;
        json_SetStr    (pJson : 'customerReference' : '');
        json_SetInt    (pJson : 'articleId'  : 0);
        json_SetStr    (pJson : 'articleDescription' : '');
        json_SetInt    (pJson : 'quantity'  : 0);
        json_SetNum    (pJson : 'articlePrice'  : 0);
        json_SetNum    (pJson : 'installationPrice'  : 0);
        endif;

        // pffakrek velden
         pfakn = json_newObject();
         json_SetNum    (pfakn : 'KEGAL'  : KEGAL);
         json_SetNum    (pfakn : 'KBDR0'  : KBDR0);
         json_SetNum    (pfakn : 'KBDR6'  : KBDR6);
         json_SetNum    (pfakn : 'KBDR17' : KBDR17);
         json_SetNum    (pfakn : 'KBDR19' : KBDR19);
         json_SetNum    (pfakn : 'KBDR25' : KBDR25);
         json_SetNum    (pfakn : 'KBDR33' : KBDR33);
         json_SetNum    (pfakn : 'KBDR0M' : KBDR0M);
         json_SetNum    (pfakn : 'KKORTB' : KKORTB);
         json_SetNum    (pfakn : 'KBD195' : KBD195);
         json_SetNum    (pfakn : 'KBDR12' : KBDR12);
         json_SetStr    (pfakn : 'KKODE'  : KKODE );
         json_SetStr    (pfakn : 'KMNP'  : KMNP );
           json_SetNum    (pfakn : 'KBDR55' : 0);
           json_SetNum    (pfakn : 'KBD196' : 0);
           json_SetNum    (pfakn : 'KBD206' : 0);
           json_SetNum    (pfakn : 'KBDR8' : 0);
         json_MoveObjectInto(pjson : 'pffakrek' : pfakn);
        // einde pffakrek velden
```

```
SOURCE FILE . . . . . . .  CDJSON/QRPGLESRC
MEMBER  . . . . . . . . .  JSONINV01

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0


 if pfirma = 'WV';
 chain (bdocnr) LFPROJFFA1;
 if %found;
 json_SetInt    (pJson : 'projectId'  : FPROJN);
 else;
 json_SetInt    (pJson : 'projectId'  : 0);
 endif;
 else;
 json_SetInt    (pJson : 'projectId'  : 0);
 endif;

 if pfirma = 'WV';
 prnr = kklnr;
 chain (prnr) lfvprs;
 if %found;
 json_SetStr    (pJson : 'CustomerEnterpriseNumber'  : PROND);
 else;
 json_SetStr    (pJson : 'CustomerEnterpriseNumber'  : '0');
 endif;
 else;
 json_SetStr    (pJson : 'CustomerEnterpriseNumber'  : '0');
 endif;

 chain (kklnr) lfklan;
 if %found;
 //json_SetNum    (pJson : 'CashDiscountPercentage'  : kprko );
 chain (kvtnr) lfvern;
 if %found;
 select;
 when vbank = 1;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'B2C');
 when vbank = 2;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'B2B');
 when vbank = 4;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'PROJ');
 when vbank = 5;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'CCH');
 when vbank = 6;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'Export');
 when vbank = 97;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : 'Franse VT');
 other;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : '');
 endsl;
 else;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : '');
 endif;
 else;
 json_SetStr    (pJson : 'BusinessUnitDimemsion' : '');
 endif;

```

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
// pfstafa velden
 setll (kfaknr) lfstafaf;
 reade (kfaknr) lfstafaf;
 dow not %eof(lfstafaf);
 // Kjeld, 2023-11-20 geen plaatsing doorsturen voor deze firma's                                   20/11/23
 if pfirma<>'WV' and pfirma<>'BA';                                                                  20/11/23
   SPLATS=0;                                                                                        20/11/23
 ENDIF;                                                                                             20/11/23
 ptemp = json_NewObject();
 json_SetInt    (ptemp : 'SSTUKS' : SSTUKS);
 json_SetNum    (ptemp : 'SHOEV'  : SHOEV);
 json_SetNum    (ptemp : 'SVERKW' : SVERKW);
 json_SetNum    (ptemp : 'SKOMMD' : SKOMMD);
 json_SetNum    (ptemp : 'SPLATS' : SPLATS);
 json_SetNum    (ptemp : 'SKOMM'  : SKOMM);
 json_SetNum    (ptemp : 'SPRIV'  : SPRIV);
 json_SetInt    (ptemp : 'SARTGR' : SARTGR);
 json_SetInt    (ptemp : 'SORDER' : SORDER);
 json_SetValue ( pJson : 'pfstafaLines[]' : ptemp);
 json_delete(pTemp);
 reade (kfaknr) lfstafaf;
 enddo;
// pfstafa velden

json_SetValue ( pNewObj : 'Invoices[]' : pjson);
json_delete(pJson);
JLIB=PFIRMA;                                                                                        22/11/23
JFAKNR=KFAKNR;                                                                                      22/11/23
JTIMST=%Timestamp();                                                                                22/11/23
write PFJSONIR;                                                                                     22/11/23
endif;
endif;
reade (kimpda) lffakinv;
enddo;
if teller > 0;
//json_WriteJsonStmf(pNewObj: '/export/inv_' + pfirma + '_' + pdatum + '.json': 1208: *OFF);        01/12/23
json_WriteJsonStmf(pNewObj: '/export/inv_' + pfirma +                                               01/12/23
           '_' + %trim(FILENAME) + '.json': 1208: *OFF);                                            01/12/23
endif;
json_delete(pNewObj);
*inlr = *on;
```

* * * * E N D O F S O U R C E * * * *