

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 lfklnr 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 /include qrpgleRef,jsonxml
1600
1700 dcl-pi *n ;
1800     pfirma char(2) ;
1900     pdatum char(8) ;
2000 end-pi ;
2100
2200 dcl-s pk1r char(1) ;
2300
2400 Dcl-S klnr zoned(8:0);
2500 Dcl-S kbase zoned(9:2);
2600 Dcl-S teller int(10);
2700 Dcl-S vatregime char(1);
2800 Dcl-S pTemp          Pointer;
2900 Dcl-S pFakn          Pointer;
3000 Dcl-S pJson          Pointer;
3100 Dcl-S pNewObj          Pointer;
3200 Dcl-S werk zoned(11:4);
3300 Dcl-S prc zoned(3:1);
3400
3500 pk1r = 'K';
3600
3700 kimpda = %dec(pdatum:8:0);
3800
3900 pNewObj = json_NewObject();
4000
4100 setll (kimpda) lffakinv;
4200 reade (kimpda) lffakinv;
4300 dow not %eof(lffakinv);
4400 chain ('001':KFAKNR) lfbew2a;
4500 if brsfla = '';
4600 if bk1r = 'K';
4700 teller = teller + 1;
4800 pJson = json_NewObject();
4900 chain (pk1r:bnum3) pfsigi;
5000 json_SetStr (pJson : 'companyId' : pfirma );
5100 json_SetInt (pJson : 'invoiceNumber' : kfaknr );
5200 //json_SetStr (pJson : 'voucherNumber' : bjn1 + '/' + %char(bdocnr) );
5300 json_SetStr (pJson : 'voucherNumber' : %char(bdocnr) );
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

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

```
5400   if kbdrf < 0;                                     16/11/23
5500   baard = 'CN';
5600   kbdrf = kbdrf * -1;
5700   kegal = kegal * -1;
5800   kbdr0 = kbdr0 * -1;
5900   kbdr6 = kbdr6 * -1;                               16/11/23
6000   kbdr17 = kbdr17 * -1;
6100   kbdr19 = kbdr19 * -1;
6200   kbdr25 = kbdr25 * -1;
6300   kbdr33 = kbdr33 * -1;
6400   kbdr0m = kbdr0m * -1;
6500   KBD195 = KBD195 * -1;
6600   KBDR12 = kbdr12 * -1;
6700   kkortb = kkortb * -1;
6800   endif;
6900
7000   json_SetStr    (pJson : 'type'   : %trim(baard) );
7100   klnr = %dec(bnum:8:0);
7200   monitor;
7300   kklnr = klnr;
7400   on-error;
7500   kklnr = 0;
7600   endmon;
7700   json_SetStr    (pJson : 'customerId' : pfirma + bnum3);
7800   json_SetNum    (pJson : 'cashDiscount' : kkortb );
7900   // herrekenen perc disc op vraag Sebastien
8000   prc = 0;
8100   if kkortb <> 0;
8200       werk = kbdrf - kbtw;
8300       werk = werk / 100;
8400   if werk <> 0;
8500       prc = %dech((kkortb / werk): 3 :1);
8600   endif;
8700   endif;
8800   json_SetNum    (pJson : 'CashDiscountPercentage' : prc );
8900   json_SetStr    (pJson : 'currency' : 'EUR' );
9000   json_SetNum    (pJson : 'totalInvoiceAmount' : kbdrf );
9100   chain (pfirma:KFAKNR) pffactur;
9200   if %found;
9300       json_SetStr    (pJson : 'OGM': %trim(KOGM));
9400   else;
9500       json_SetStr    (pJson : 'OGM': '');
9600   endif;
9700   if bvald = *blanks;
9800       eval bvald = bimpd;
9900   endif;
10000  json_SetStr    (pJson : 'invoiceDate' : %subst(bimpd:1:4) + '-' +
10100                                %subst(bimpd:5:2) + '-' + %subst(bimpd:7:2));
10200  json_SetStr    (pJson : 'dueDate' : %subst(bvald:1:4) + '-' +
10300                                %subst(bvald:5:2) + '-' + %subst(bvald:7:2));
10400  json_SetNum    (pJson : 'totalVatAmount' : kbtw );
10500  kbase = kbdrf - kbtw - kkortb;
10600  json_SetNum    (pJson : 'totalBaseAmount' : kbase );
```

SOURCE FILE CDJSON/QRPGLSRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10700  json_SetStr      (pJson : 'vatCode' : bbtwc );
10800  json_SetStr      (pJson : 'description' : boms );
10900  json_SetStr      (pJson : 'invoiceCountry' : %trimr(LANDK));
11000  json_SetStr      (pJson : 'deliveryCountry' : %trimr(LANDK));
11100  json_SetInt       (pJson : 'orderNumber' : bordnr);
11200
11300  // check VAT regime
11400  // G = IC goederen
11500  // E = Export
11600  // S = IC diensten
11700  // M = medecontractant
11800  vatregime = *blanks;
11900  chain (bjnl:bdocnr:bimpd:bk1r3:bnun3:'R':'700014') lfbew2a;
12000  if %found;
12100  vatregime = 'S';
12200  endif;
12300  chain (bjnl:bdocnr:bimpd:bk1r3:bnun3:'R':'700015') lfbew2a;
12400  if %found;
12500  vatregime = 'M';
12600  endif;
12700  chain (bjnl:bdocnr:bimpd:bk1r3:bnun3:'R':'700016') lfbew2a;
12800  if %found;
12900  vatregime = 'G';
13000  endif;
13100  chain (bjnl:bdocnr:bimpd:bk1r3:bnun3:'R':'700017') lfbew2a;
13200  if %found;
13300  vatregime = 'E';
13400  endif;
13500  chain (bjnl:bdocnr:bimpd:bk1r3:bnun3:'R':'700110') lfbew2a;
13600  if %found;
13700  vatregime = 'G';
13800  endif;
13900
14000  if vatregime = *blanks;
14100  if pfirma = 'WV';
14200  chain (bdocnr) PFVERKOP;
14300  if %found;
14400  vatregime = 'E';
14500  endif;
14600  endif;
14700  if kbdr0m <> 0;
14800  vatregime = 'M';
14900  endif;
15000  endif;
15100
15200  if kbdrf = 0;
15300  vatregime = 'Z';
15400  endif;
15500
15600  json_SetStr      (pJson : 'vatRegime' : %trimr(vatregime));
15700
15800  chain (pfirma:bdocnr) vatlogf;
15900  if %found;
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
16000  vregime = vatregime;
16100  update vatr;
16200  else;
16300  vfirma = pfirma;
16400  vdocnr = bdocnr;
16500  vregime = vatregime;
16600  write vatr;
16700  endif;
16800
16900  chain bordnr lforderb;
17000  if %found;
17100  json_SetStr      (pJson : 'customerReference' : %trimr(OREFK));
17200  json_SetInt      (pJson : 'articleId' : oartnr);
17300  json_SetStr      (pJson : 'articleDescription' : %trimr(OARTUI));
17400  json_SetInt      (pJson : 'quantity' : ostuks);
17500  json_SetNum      (pJson : 'articlePrice' : ofaktw);
17600  json_SetNum      (pJson : 'installationPrice' : owerpl);
17700  else;
17800  json_SetStr      (pJson : 'customerReference' : '');
17900  json_SetInt      (pJson : 'articleId' : 0);
18000  json_SetStr      (pJson : 'articleDescription' : '');
18100  json_SetInt      (pJson : 'quantity' : 0);
18200  json_SetNum      (pJson : 'articlePrice' : 0);
18300  json_SetNum      (pJson : 'installationPrice' : 0);
18400  endif;
18500
18600  // pffakrek velden
18700  pfakn = json_newObject();
18800  json_SetNum      (pfakn : 'KEGAL' : KEGAL);
18900  json_SetNum      (pfakn : 'KBDR0' : KBDR0);
19000  json_SetNum      (pfakn : 'KBDR6' : KBDR6);
19100  json_SetNum      (pfakn : 'KBDR17' : KBDR17);
19200  json_SetNum      (pfakn : 'KBDR19' : KBDR19);
19300  json_SetNum      (pfakn : 'KBDR25' : KBDR25);
19400  json_SetNum      (pfakn : 'KBDR33' : KBDR33);
19500  json_SetNum      (pfakn : 'KBDR0M' : KBDR0M);
19600  json_SetNum      (pfakn : 'KKORTB' : KKORTB);
19700  json_SetNum      (pfakn : 'KBD195' : KBD195);
19800  json_SetNum      (pfakn : 'KBDR12' : KBDR12);
19900  json_SetStr      (pfakn : 'KKODE' : KKODE );
20000  json_SetStr      (pfakn : 'KMNP' : KMNP );
20100  json_SetNum      (pfakn : 'KBDR55' : 0);
20200  json_SetNum      (pfakn : 'KBD196' : 0);
20300  json_SetNum      (pfakn : 'KBD206' : 0);
20400  json_SetNum      (pfakn : 'KBDR8' : 0);
20500  json_MoveObjectInto(pjson : 'pffakrek' : pfakn);
20600  // einde pffakrek velden
20700
20800
20900  if pfirma = 'WV';
21000  chain (bdocnr) LFPROJFFA1;
21100  if %found;
21200  json_SetInt      (pJson : 'projectId' : FPROJN);
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
21300  else;
21400  json_SetInt      (pJson : 'projectId' : 0);
21500  endif;
21600  else;
21700  json_SetInt      (pJson : 'projectId' : 0);
21800  endif;
21900
22000  if pfirma = 'WV';
22100  prnr = kklmr;
22200  chain (prnr) lfvprs;
22300  if %found;
22400  json_SetStr      (pJson : 'CustomerEnterpriseNumber' : PROND);
22500  else;
22600  json_SetStr      (pJson : 'CustomerEnterpriseNumber' : '0');
22700  endif;
22800  else;
22900  json_SetStr      (pJson : 'CustomerEnterpriseNumber' : '0');
23000  endif;
23100
23200  chain (kklmr) lfklnr;
23300  if %found;
23400  //json_SetNum      (pJson : 'CashDiscountPercentage' : kprko );
23500  chain (kvtmr) lfvprn;
23600  if %found;
23700  select;
23800  when vbank = 1;
23900  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'B2C');
24000  when vbank = 2;
24100  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'B2B');
24200  when vbank = 4;
24300  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'PROJ');
24400  when vbank = 5;
24500  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'CCH');
24600  when vbank = 6;
24700  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'Export');
24800  when vbank = 97;
24900  json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'Franse VT');
25000  other;
25100  json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
25200  endl;
25300  else;
25400  json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
25500  endif;
25600  else;
25700  json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
25800  endif;
25900
26000  // pfstafa velden
26100  setll (kfakmr) lfstafaf;
26200  reade (kfakmr) lfstafaf;
26300  dow not %eof(lfstafaf);
26301  // Kjeld, 2023-11-20 geen plaatsing doorsturen voor deze firma's
26302  if pfirma<>'WV' and pfirma<>'BA';
```

20/11/23

20/11/23

SOURCE FILE CDJSON/QRPGLSRC

MEMBER JSONINV01

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

26303 SPLATS=0;

20/11/23

26304 ENDIF;

20/11/23

26400 ptemp = json_NewObject();

26500 json_SetInt (ptemp : 'SSTUKS' : SSTUKS);

26600 json_SetNum (ptemp : 'SHOEV' : SHOEV);

26700 json_SetNum (ptemp : 'SVERKW' : SVERKW);

26800 json_SetNum (ptemp : 'SKOMMD' : SKOMMD);

26900 json_SetNum (ptemp : 'SPLATS' : SPLATS);

27000 json_SetNum (ptemp : 'SKOMM' : SKOMM);

27100 json_SetNum (ptemp : 'SPRIV' : SPRIV);

27200 json_SetInt (ptemp : 'SARTGR' : SARTGR);

27300 json_SetInt (ptemp : 'SORDER' : SORDER);

27400 json_SetValue (pJson : 'pfstafaLines[]' : ptemp);

27500 json_delete(pTemp);

27600 reade (kfaknr) lfstafaf;

27700 enddo;

27800 // pfstafa velden

27900

28000 json_SetValue (pNewObj : 'Invoices[]' : pjson);

28100 json_delete(pJson);

28200 endif;

28300 endif;

28400 reade (kimpda) lffakinv;

28500 enddo;

28600 if teller > 0;

28700 json_WriteJsonStmf(pNewObj: '/export/inv_' + pfirma + '_' + pdatum + '.json': 1208: *OFF);

28800 endif;

28900 json_delete(pNewObj);

29000 *inlr = *on;

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