

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 lfkln 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;
1600 /include qrpgleRef,jsonxml
1700
1800 dcl-pi *n ;
1900     pfirma char(2) ;
2000     pdatum char(8) ;
2100 end-pi ;
2200
2300 dcl-s pk1r 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);
3402 FILENAME = %trim(%char(%timestamp(*sys:0)));
3500
3600 pk1r = '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;
4600 if not %found;
4700 chain ('001':KFAKNR) lfbew2a;
4800 //if bbuf1a = '';
4900 if bk1r = 'K';
5000 teller = teller + 1;
5100 pJson = json_NewObject();
```

27/11/23

01/12/23

01/12/23

27/11/23

27/11/23

27/11/23

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

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

```

5200 chain (pklr:bnum3) pfsigi;
5300 json_SetStr      (pJson : 'companyId' : pfirma );
5400 json_SetInt      (pJson : 'invoiceNumber' : kfaknr );
5500 //json_SetStr     (pJson : 'voucherNumber' : bjn1 + '/' + %char(bdocnr) );
5600 json_SetStr      (pJson : 'voucherNumber' : %char(bdocnr) );
5700 if kbdrf < 0;
5800 baard = 'CN';
5900 kbdrf = kbdrf * -1;
6000 kbtw = kbtw * -1;
6100 kegal = kegal * -1;
6200 kbdr0 = kbdr0 * -1;
6300 kbdr6 = kbdr6 * -1;
6400 kbdr17 = kbdr17 * -1;
6500 kbdr19 = kbdr19 * -1;
6600 kbdr25 = kbdr25 * -1;
6700 kbdr33 = kbdr33 * -1;
6800 kbdr0m = kbdr0m * -1;
6900 KBD195 = KBD195 * -1;
7000 KBD12 = kbdr12 * -1;
7100 kkortb = kkortb * -1;
7200 endif;
7300
7400 json_SetStr      (pJson : 'type' : %trim(baard) );
7500 klmr = %dec(bnum:8:0);
7600 monitor;
7700 kklmr = klmr;
7800 on-error;
7900 kklmr = 0;
8000 endmon;
8100 json_SetStr      (pJson : 'customerId' : pfirma + bnum3);
8200 json_SetNum      (pJson : 'cashDiscount' : kkortb );
8300 // herrekenen perc disc op vraag Sebastien
8400 prc = 0;
8500 if kkortb <> 0;
8600 werk = kbdrf - kbtw;
8700 werk = werk / 100;
8800 if werk <> 0;
8900 prc = %dech((kkortb / werk): 3 :1);
9000 endif;
9100 endif;
9200 json_SetNum      (pJson : 'CashDiscountPercentage' : prc );
9300 json_SetStr      (pJson : 'currency' : 'EUR' );
9400 json_SetNum      (pJson : 'totalInvoiceAmount' : kbdrf );
9500 chain (pfirma:KFAKNR) pffactur;
9600 if %found;
9700 json_SetStr      (pJson : 'OGM': %trim(KOGM));
9800 else;
9900 json_SetStr      (pJson : 'OGM': '');
10000 endif;
10100 if bvald = *blanks;
10200 eval bvald = bimpd;
10300 endif;
10400 json_SetStr      (pJson : 'invoiceDate' : %subst(bimpd:1:4) + '-' +

```

SOURCE FILE CDJSON/QRPGLSRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10500      %subst(bimpd:5:2) + '-' + %subst(bimpd:7:2));
10600      json_SetStr      (pJson : 'dueDate' : %subst(bvald:1:4) + '-' +
10700      %subst(bvald:5:2) + '-' + %subst(bvald:7:2));
10800      json_SetNum      (pJson : 'totalVatAmount' : kbtw );
10900      kbase = kbdrf - kbtw - kkortb;
11000      json_SetNum      (pJson : 'totalBaseAmount' : kbase );
11100      json_SetStr      (pJson : 'vatCode' : bbtwc );
11200      json_SetStr      (pJson : 'description' : boms );
11300      json_SetStr      (pJson : 'invoiceCountry' : %trimr(LANDK));
11400      json_SetStr      (pJson : 'deliveryCountry' : %trimr(LANDK));
11500      json_SetInt      (pJson : 'orderNumber' : bordnr);
11600
11700      // check VAT regime
11800      // G = IC goederen
11900      // E = Export
12000      // S = IC diensten
12100      // M = medecontractant
12200      vatregime = *blanks;
12300      chain (bjnl:bdocnr:bimpd:bk1r3:bnum3:'R':'700014') lfbew2a;
12400      if %found;
12500      vatregime = 'S';
12600      endif;
12700      chain (bjnl:bdocnr:bimpd:bk1r3:bnum3:'R':'700015') lfbew2a;
12800      if %found;
12900      vatregime = 'M';
13000      endif;
13100      chain (bjnl:bdocnr:bimpd:bk1r3:bnum3:'R':'700016') lfbew2a;
13200      if %found;
13300      vatregime = 'G';
13400      endif;
13500      chain (bjnl:bdocnr:bimpd:bk1r3:bnum3:'R':'700017') lfbew2a;
13600      if %found;
13700      vatregime = 'E';
13800      endif;
13900      chain (bjnl:bdocnr:bimpd:bk1r3:bnum3:'R':'700110') lfbew2a;
14000      if %found;
14100      vatregime = 'G';
14200      endif;
14300
14400      if vatregime = *blanks;
14500      if pfirma = 'WV';
14600      chain (bdocnr) PFVERKOP;
14700      if %found;
14800      vatregime = 'E';
14900      endif;
15000      endif;
15100      if kbdr0m <> 0;
15200      vatregime = 'M';
15300      endif;
15400      endif;
15500
15600      if kbdrf = 0;
15700      vatregime = 'Z';
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
15800   endif;
15900
16000   json_SetStr      (pJson : 'vatRegime' : %trimr(vatregime));
16100
16200   chain (pfirma:bdocnr) vatlogf;
16300   if %found;
16400     vregime = vatregime;
16500     update vatr;
16600   else;
16700     vfirma = pfirma;
16800     vdocnr = bdocnr;
16900     vregime = vatregime;
17000     write vatr;
17100   endif;
17200
17300   chain bordnr lfororderb;
17400   if %found;
17500     json_SetStr      (pJson : 'customerReference' : %trimr(OREFK));
17600     json_SetInt      (pJson : 'articleId' : oartnr);
17700     json_SetStr      (pJson : 'articleDescription' : %trimr(OARTUI));
17800     json_SetInt      (pJson : 'quantity' : ostuks);
17900     json_SetNum      (pJson : 'articlePrice' : ofaktw);
18000     json_SetNum      (pJson : 'installationPrice' : owerpl);
18100   else;
18200     json_SetStr      (pJson : 'customerReference' : '');
18300     json_SetInt      (pJson : 'articleId' : 0);
18400     json_SetStr      (pJson : 'articleDescription' : '');
18500     json_SetInt      (pJson : 'quantity' : 0);
18600     json_SetNum      (pJson : 'articlePrice' : 0);
18700     json_SetNum      (pJson : 'installationPrice' : 0);
18800   endif;
18900
19000   // pffakrek velden
19100   pfakn = json_newObject();
19200   json_SetNum      (pfakn : 'KEGAL' : KEGAL);
19300   json_SetNum      (pfakn : 'KBDR0' : KBDR0);
19400   json_SetNum      (pfakn : 'KBDR6' : KBDR6);
19500   json_SetNum      (pfakn : 'KBDR17' : KBDR17);
19600   json_SetNum      (pfakn : 'KBDR19' : KBDR19);
19700   json_SetNum      (pfakn : 'KBDR25' : KBDR25);
19800   json_SetNum      (pfakn : 'KBDR33' : KBDR33);
19900   json_SetNum      (pfakn : 'KBDR0M' : KBDR0M);
20000   json_SetNum      (pfakn : 'KKORTB' : KKORTB);
20100   json_SetNum      (pfakn : 'KBD195' : KBD195);
20200   json_SetNum      (pfakn : 'KBDR12' : KBDR12);
20300   json_SetStr      (pfakn : 'KKODE' : KKODE );
20400   json_SetStr      (pfakn : 'KMNP' : KMNP );
20500   json_SetNum      (pfakn : 'KBDR55' : 0);
20600   json_SetNum      (pfakn : 'KBD196' : 0);
20700   json_SetNum      (pfakn : 'KBD206' : 0);
20800   json_SetNum      (pfakn : 'KBDR8' : 0);
20900   json_MoveObjectInto(pjson : 'pffakrek' : pfakn);
21000   // einde pffakrek velden
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

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

```
21100
21200
21300     if pfirma = 'WV';
21400     chain (bdocnr) LFPROJFFA1;
21500     if %found;
21600     json_SetInt      (pJson : 'projectId'   : FPROJN);
21700     else;
21800     json_SetInt      (pJson : 'projectId'   : 0);
21900     endif;
22000     else;
22100     json_SetInt      (pJson : 'projectId'   : 0);
22200     endif;
22300
22400     if pfirma = 'WV';
22500     prnr = kklnr;
22600     chain (prnr) lfvprs;
22700     if %found;
22800     json_SetStr      (pJson : 'CustomerEnterpriseNumber' : PROND);
22900     else;
23000     json_SetStr      (pJson : 'CustomerEnterpriseNumber' : '0');
23100     endif;
23200     else;
23300     json_SetStr      (pJson : 'CustomerEnterpriseNumber' : '0');
23400     endif;
23500
23600     chain (kklnr) lfkln;
23700     if %found;
23800     //json_SetNum      (pJson : 'CashDiscountPercentage' : kprko );
23900     chain (kvtnr) lfvtrn;
24000     if %found;
24100     select;
24200     when vbank = 1;
24300     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'B2C');
24400     when vbank = 2;
24500     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'B2B');
24600     when vbank = 4;
24700     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'PROJ');
24800     when vbank = 5;
24900     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'CCH');
25000     when vbank = 6;
25100     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'Export');
25200     when vbank = 97;
25300     json_SetStr      (pJson : 'BusinessUnitDimemsion' : 'Franse VT');
25400     other;
25500     json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
25600     ends1;
25700     else;
25800     json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
25900     endif;
26000     else;
26100     json_SetStr      (pJson : 'BusinessUnitDimemsion' : '');
26200     endif;
26300
```

SOURCE FILE CDJSON/QRPGLESRC

MEMBER JSONINV01

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
26400 // pfstafa velden
26500 setll (kfaknr) lfstafaf;
26600 reade (kfaknr) lfstafaf;
26700 dow not %eof(lfstafaf);
26800 // Kjeld, 2023-11-20 geen plaatsing doorsturen voor deze firma's
26900 if pfirma<>'WV' and pfirma<>'BA';
27000 SPLATS=0;
27100 ENDIF;
27200 ptemp = json_NewObject();
27300 json_SetInt (ptemp : 'SSTUKS' : SSTUKS);
27400 json_SetNum (ptemp : 'SHOEV' : SHOEV);
27500 json_SetNum (ptemp : 'SVERKW' : SVERKW);
27600 json_SetNum (ptemp : 'SKOMMD' : SKOMMD);
27700 json_SetNum (ptemp : 'SPLATS' : SPLATS);
27800 json_SetNum (ptemp : 'SKOMM' : SKOMM);
27900 json_SetNum (ptemp : 'SPRIV' : SPRIV);
28000 json_SetInt (ptemp : 'SARTGR' : SARTGR);
28100 json_SetInt (ptemp : 'SORDER' : SORDER);
28200 json_SetValue ( pJson : 'pfstafaLines[]' : ptemp);
28300 json_delete(pTemp);
28400 reade (kfaknr) lfstafaf;
28500 enddo;
28600 // pfstafa velden
28700
28800 json_SetValue ( pNewObj : 'Invoices[]' : pjson);
28900 json_delete(pJson);
29000 JLIB=PFIRMA;
29100 JFAKNR=KFAKNR;
29200 JTIMST=%Timestamp();
29300 write PFJSONIR;
29400 endif;
29500 endif;
29600 reade (kimpda) lffakinv;
29700 enddo;
29800 if teller > 0;
29801 //json_WriteJsonStmf(pNewObj: '/export/inv_' + pfirma + '_' + pdatum + '.json': 1208: *OFF);
29900 json_WriteJsonStmf(pNewObj: '/export/inv_' + pfirma +
29901 '_' + %trim(FILENAME) + '.json': 1208: *OFF);
30000 endif;
30100 json_delete(pNewObj);
30200 *inlr = *on;
```

20/11/23

20/11/23

20/11/23

20/11/23

22/11/23

22/11/23

22/11/23

22/11/23

01/12/23

01/12/23

01/12/23

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