

MT2ASCII

MSX Computer & Club Magazine nummer 69 - augustus 1994

Arno Dekker

Scanned, ocr'ed and converted to PDF by HansO, 2001

In de laatste serie MSX Club Magazine's werd at dieper ingegaan op MT-Base files en er werd gesproken over het maken van een converter van MT-Base naar ASCII, waarop positieve reacties kwamen.

Die converter is er nu.

Lange tijd bleef het stil rond de converter. Uw hoofdredacteur meende zelf zoiets snel in elkaar te kunnen draaien, maar had geen goede files ter beschikking om een en ander te testen en daarom bleef het liggen. Ik zend nu een programmaatje in dat ik in het begin van dit jaar heb gemaakt. Het programma is geschreven in Turbo Pascal en zet MT-Base bestanden om in ASCII tekstfiles.

Opstarten gaat onder MSXDOS door MT2ASCII in te typen. Het snelst werkt het programma uiteraard met een RAM-disk, maar de snelheid bij gebruik van de gewone diskette is ook nog wel aanvaardbaar, zeker bij kleinere bestanden.

Werking

Er wordt gecontroleerd of de om te zetten MT-Base file op de disk staat en of het ook daadwerkelijk een MT-Base file is. Verder wordt gecontroleerd of de naam van de aan te maken ASCII file al bestaat. Voor de bestandsnaam kan de drive opgegeven worden, anders wordt uitgegaan van de default drive.

Problemen

Het enige probleem dat tot nu toe niet verholpen is, is dat de omzetting verkeerd gaat, indien bij het declareren van de velden er velden worden overgeslagen. In de praktijk zal dit gelukkig weinig gebeuren. Verder ben ik geen problemen tegengekomen, maar misschien zijn er nog meer situaties te bedenken waarbij het fout kan gaan.

Houd er rekening mee, dat, als er veel of lange velden zijn gedeclareerd, de ASCII-file regels langer worden dan 80 kolommen, maar door in een smaller / kleiner lettertype af te drukken hoeft dit geen problemen op te leveren.

Tenslotte

Het programmaatje staat in de disk image inclusief de sourcefile in Turbo Pascal aan het eind van het artikel, zodat u het zelf kunt aanpassen. Kijk niet teveel naar deze source; ik vind het niet altijd een modelvoorbeeld van gestructureerd programmeren. Op- en aanmerkingen zie ik echter gaarne tegemoet.

```

PROGRAM mt2ascii (INPUT,OUTPUT);
{Dit programma zet een MT-BASE file om in een ASCII(tekst)-file}

TYPE
string15=STRING[15];
string20=STRING[20];
string100=STRING[100];
plaats=ARRAY[1..45] OF INTEGER;

VAR
oud,nieuw:TEXT;
g:plaats;
kar,karl:CHAR;
foutmelding:string100;
c,h,io,teller,n,y,blok:INTEGER;
sizeblok:REAL;

PROCEDURE kop;
VAR naam:STRING[13];
    lengte,totaal:INTEGER;
{Nu worden twee files geopend. Er wordt gekeken of de file die moet
worden omgezet al op schijf staat. Zo niet, stopt het programma,
net als wanneer de aan te maken file juist wel bestaat}
BEGIN {kop}
    CLRSCR;
    WRITELN;
    WRITELN('=====');
    WRITELN('MT-BASE to ASCII Converter v1.0');
    WRITE('Made by ');
    naam:='Arno Dekker';
    lengte:=LENGTH(naam);
    y:=1;
    totaal:=0;
    WHILE lengte<>0 DO
    BEGIN
        totaal:=totaal+(ORD(COPY(naam,y,1)));
        y:=y+1;
        lengte:=lengte-1
    END;
    WHILE totaal<>1094 DO
    BEGIN
        WRITELN('Wil je ergens je naam bij hebben staan, maak dan zelf
wat!');
        END;
        WRITELN (naam,' 1994');
        WRITELN('This program is Public Domain');
        WRITELN('=====')
    END; {kop}

PROCEDURE openfiles;
VAR oudenaam,nieuwenaam:string15;
BEGIN {openfiles}
    foutmelding:=' ';
    io:=0;
    WRITELN;
    WRITE('Naam MT-BASE file (zonder ext.):');
    READLN(oudenaam);
    oudenaam:=oudenaam+'.MTF';
    WRITE('Naam nieuw te maken ASCII-file :');
    READLN(nieuwenaam);

```

```

ASSIGN(oud,oudenaam);
{$I-} RESET(oud); {$I+} {open file voor lezen}
IF ioresult=1 THEN      {oude file niet op disk}
BEGIN
  io:=1;
  WRITELN;
  foutmelding:='Deze MT-BASE file is niet op de diskette
aanwezig!';
END;
{Controle 'M' op positie $00E (eerste sector). Ja->MT-BASE file}
teller:=0;
WHILE (teller<>3) AND (io=0) DO
BEGIN
  READ(oud,kar);
  teller:=teller+1
END; {read}
IF (kar<>CHR(61)) AND (io=0) THEN
BEGIN {geen MT}
  foutmelding:='Dit is geen MT-BASE bestand!';
  io:=1;
END; {geen MT}
IF (kar=CHR(61)) AND (io=0) THEN
BEGIN {wel MT}
  WRITELN('Herkent als MT-BASE file...');
  teller:=0;
  WHILE teller<> 30 DO
  BEGIN {read}
    READ(oud,kar);
    teller:=teller+1
  END; {read}
END; {Wel MT}
ASSIGN(nieuw,nieuwenaam);
{$I-} RESET(nieuw); {$I+}
IF ioresult<>1 THEN      {nieuwe file bestaat al}
BEGIN {bestaat}
  io:=1;
  WRITELN;
  foutmelding:='De naam van de te maken ASCII file bestaat al!';
END; {bestaat}
IF io=0 THEN
  REWRITE(nieuw);      {open file voor schrijven}
END; {openfiles}

PROCEDURE VNAMEN;
{Lees vanaf positie $021 (eerste sector) 12 posities uit, dit
totdat eerste positie '20' (spatie) bevat, dan als een ASCII
regel wegschrijven, of anders indien het max van 14 veldnamen
is bereikt. (Het aantal veldnamen onthouden).}
BEGIN
  teller:=0;
  y:=14;
  n:=0;
  READ(oud,kar);
  WRITE(nieuw,kar);
  WHILE y<>n DO
  BEGIN
    WHILE teller<> 11 DO
    BEGIN
      READ(oud,kar);
      WRITE(nieuw,kar);
      teller:=teller+1;
    END;
  END;

```

```

END; {teller<>11,read & write}
n:=n+1; {aantal velden}
WRITE(nieuw,CHR($20)); {spatie ertussen}
READ(oud,kar);
IF kar=CHR($20) THEN
BEGIN
WRITE(nieuw,CHR(13)+CHR(10)); {return}
y:=n; {geen velden meer?}
END; {return}
IF kar<>CHR($20) THEN WRITE(nieuw,kar);
teller:=0;
END; {y<>n}
teller:=n*12;
WHILE teller<>0 DO
BEGIN
WRITE(nieuw,'-');
teller:=teller-1;
END;
WRITE(nieuw,CHR(13)+CHR(10)); {return}
END; {procedure}

PROCEDURE recinfo;
{Lees vanaf positie $13D veldnaamaantal posities in (max 14,
dus positie $14B max bereikt). Onthoud deze waarden voor
plaatsen data van MT-BASE op goede plaats in ASCII bestand.
Het totaal aantal records wordt gelezen op posities $178
en $179 van de eerste sector.}
BEGIN
WRITELN;
WRITELN('Aantal gevonden velden      : ',n);
teller:=$13D-($20+(n*12+1));
y:=0;
WHILE y<>teller DO
BEGIN
READ(oud,kar);
y:=y+1;
END; {naar $13D}
h:=1;
teller:=0;
c:=0;
WHILE (h<15) AND (teller=0) DO
BEGIN
READ(oud,kar);
g[h]:=ORD(kar);
IF g[h]=c THEN teller:=1;
c:=g[h];
h:=h+1
END;
IF h=15 THEN h:=h-1 ELSE h:=h-2;
c:=h; {aantal velden}
sizeblok:=510/g[h];
blok:=TRUNC(sizeblok);
WHILE c<>1 DO
BEGIN
g[c]:=g[c]-g[c-1];
c:=c-1;
END;
teller:=0;
IF h=14 THEN h:=h-1;
WHILE teller<>(45+(13-(h+1))) DO
BEGIN

```

```

    READ(oud,kar);
    teller:=teller+1
END;
READ(oud,kar);
kar1:=kar;
READ(oud,kar);
y:=ORD(kar)*$100;
io:=ORD(kar1);
teller:=y+io;
WRITELN('Aantal records te verwerken: ',teller);
WRITELN('Blokgrootte van records      : ',blok);
WRITELN('Aantal records nog te gaan : ',teller);
y:=0;
WHILE y<>1021 DO {Naar data toe}
BEGIN
    READ(oud,kar);
    y:=y+1;
END;{overslaan}
END;{procedure}

PROCEDURE gegevens;
{Vanaf derde sector, positie $177 staat de data. Lees nu
het aantal posities uit gevonden op $13E, dan aantal van
$13F etc. t/m aantal veldnamen. Dit alles moet het totaal
aantal record keer. Dit alles moet worden weggeschreven
op dezelfde posities als ze in MT-BASE zijn genoteerd.}
VAR tel,kaart:INTEGER;
BEGIN
    tel:=1;
    kaart:=0;
    WHILE (teller<>0) AND (NOT EOF(oud)) DO
    BEGIN {totverw}
        io:=0;
        IF kaart = blok THEN
        BEGIN
            READ(oud,kar);
            READ(oud,kar);
            READ(oud,kar);
            kaart:=0;
        END;
        c:=n; {aantal gevonden velden}
        WHILE c <> 0 DO
        BEGIN {record verwerken}
            io:=g[tel];
            WHILE io <> 0 DO
            BEGIN {veld verwerken}
                READ (oud,kar);
                WRITE(nieuw,kar);
                io:=io-1
            END; {veld verwerken}
            c:=c-1;
            IF c>=1 THEN WRITE(nieuw,CHR($20));
            tel:=tel+1;
        END; {record verwerken}
        WRITE(nieuw,CHR(13)+CHR(10));
        tel:=1;
        teller:=teller-1;
        kaart:=kaart+1;
        GOTOXY(30,15);
        WRITE(teller);
        WRITE('  ')

```

```
END; {totverw}
WRITELN;
WRITELN('Alle records zijn verwerkt....');
io:=0
END; {procedure}
```

```
{Hoofdprogramma}
BEGIN {hoofdprog}
  KOP;
  OPENFILES;
  IF io=0 THEN
  BEGIN {verwerk}
    VNAMEN;
    RECINFO;
    GEGEVENS
  END; {verwerk}
  IF io<>0 THEN WRITELN(foutmelding);
  CLOSE(oud);
  CLOSE(nieuw)
END. {hoofdprog}
```